A PAIRED-ALGORITHM CLUSTERING MODEL FOR DESCRIBING FIELD STAFF DEPLOYMENT IN NON-GOVERNMENTAL ORGANIZATIONS (NGOS).

BY

MANASSES N. NYAKADO

MASTER OF SCIENCE IN DATA ANALYTICS

KCA UNIVERSITY

2025

A PAIRED-ALGORITHM CLUSTERING MODEL FOR DESCRIBING FIELD STAFF DEPLOYMENT IN NON-GOVERNMENTAL ORGANIZATIONS (NGOS).

BY

MANASSES N. NYAKADO

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE IN DATA ANALYTICS IN THE SCHOOL OF TECHNOLOGY AT KCA UNIVERSITY

MAY, 2025

DECLARATION

I declare that this dissertation is my original work and has not been previously published or submitted elsewhere for award of a degree. I also declare that this contains no material written or published by other people except where due reference is made and author duly acknowledged.

Student Name: Manasses Ndonga Nyakado Reg,No. 21/02612

Sign: _____ Date: 19 May 2025

I do hereby confirm that I have examined the master's dissertation of Manasses Ndonga Nyakado And have certified that all revisions that the dissertation panel and examiners recommended have been adequately addressed.

Name:
Position:
Signature:
Date:

Simon Mwendia

Digitally signed by Simon Mwendia
Date: 2025.05.18 15:29:15 +03'00'

Dr. Simon Mwendia
Dissertation Supervisor

A PAIRED-ALGORITHM CLUSTERING MODEL FOR DESCRIBING FIELD STAFF DEPLOYMENT IN NON-GOVERNMENTAL ORGANIZATIONS (NGOS).

ABSTRACT

This research addresses the inefficiencies and challenges faced by non-governmental organizations (NGOs) in deploying field staff, focusing on the manual processes prevalent in the current systems and leveraging on the possibilities offered by predictive machine learning algorithms. The problem stems from time-consuming and error-prone manual data entry methods, hindering optimal resource allocation. Our objective is to develop and implement a machine learning clustering algorithm to automate the field staff deployment process. By leveraging data analytics – hierarchical and k-means machine learning algorithms – we aim to enhance the efficiency and accuracy of deployment, leading to improved allocation of personnel and resources. The expected outcome is a streamlined deployment system that significantly reduces errors, minimizes time consumption, and maximizes overall operational efficiency in NGO field operations. The project outcomes will also inform advances in the use of combined methods in clustering machine learning algorithms and data analytics.

Keywords: *Combined Machine learning clustering algorithm, Machine Learning in NGOs, Hierarchical Clustering, K-means clustering, NGOs, Staff Deployment.*

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I extend appreciation to my supervisor, Dr. Simon Mwendia, for setting aside time and expertise to guide me through every step of this project – including swift responses to my late-night emails. His expertise, experience, invaluable insights, helpful feedback, and unwavering support reflect the success of this dissertation.

I am grateful to my family as well. Thank you for always being there for me and for believing in me through the high and lows of this process.

LIST OF FIGURES

GLOSSARY

1. Field staff deployment: The process of allocating and assigning personnel to specific tasks, projects, or locations.
2. Non-governmental organizations (NGOs): Private, non-profit organizations that operate independently of government and are typically concerned with social or environmental issues.
3. Data pre-processing: The process of cleaning, transforming, and organizing raw data before it can be used for analysis.
4. Feature engineering: The process of selecting, extracting, and transforming relevant features from raw data to improve machine learning algorithm performance.
5. Algorithm: A set of instructions used to solve a specific problem or perform a specific task, often used in machine learning algorithms to make predictions.
6. Performance evaluation: The process of assessing the effectiveness and accuracy of a machine learning algorithm.
7. Machine learning: A field of study that focuses on developing algorithms and statistical algorithms that enable computer systems to learn from data and make predictions or decisions based on that data.
8. Clustering: A technique in machine learning that involves grouping similar data points based on certain characteristics or features.
9. Deployment: The process of assigning or distributing personnel or resources to specific locations or tasks.
10. Non-governmental organization (NGO): A non-profit organization that operates independently of the government and aims to address a particular social or humanitarian issue.
11. Python: A high-level programming language commonly used for data analysis, machine learning, and scientific computing.
12. Heroku: A cloud platform that allows developers to build, deploy, and manage web applications.
13. Feature engineering: The process of selecting and transforming relevant data features to improve the accuracy and performance of a machine learning algorithm.
14. Supervised learning: A type of machine learning where the algorithm is trained using labeled data (i.e., data where the correct output is known) to predict outcomes for new, unlabeled data.
15. Unsupervised learning: A type of machine learning where the algorithm is trained using unlabeled data to identify patterns or groupings in the data.
16. Inefficiency**:** A state or condition where resources, such as time, labor, or materials, are not used in the best possible manner, leading to waste and suboptimal outcomes. In the context of NGO operations, inefficiency can result in delayed responses, increased operational costs, and reduced impact on target communities.
17. Underutilization**:** The condition of not using resources, such as staff or equipment, to their full capacity. This can lead to wasted potential and missed opportunities for productivity and effectiveness. In NGO field staff deployment, underutilization might mean having qualified staff members idle or not engaged in tasks where their skills are most needed.
18. Misallocation**:** The incorrect or inappropriate assignment of resources, such as funds, personnel, or materials, resulting in inefficiencies and reduced effectiveness. In the context of NGO operations, misallocation can occur when staff are deployed to areas where their skills are not needed, or resources are distributed to regions that do not require them, leading to suboptimal outcomes.

ACRONYMS

1. NGO: Non-Governmental Organization
2. CRM: Customer Relationship Management
3. ERP: Enterprise Resource Planning
4. HRM: Human Resource Management
5. GPS: Global Positioning System
6. API: Application Programming Interface
7. PaaS: Platform-as-a-Service
8. DBSCAN: Density-Based Spatial Clustering of Applications with Noise

CHAPTER ONE

1. INTRODUCTION

1.1 Background

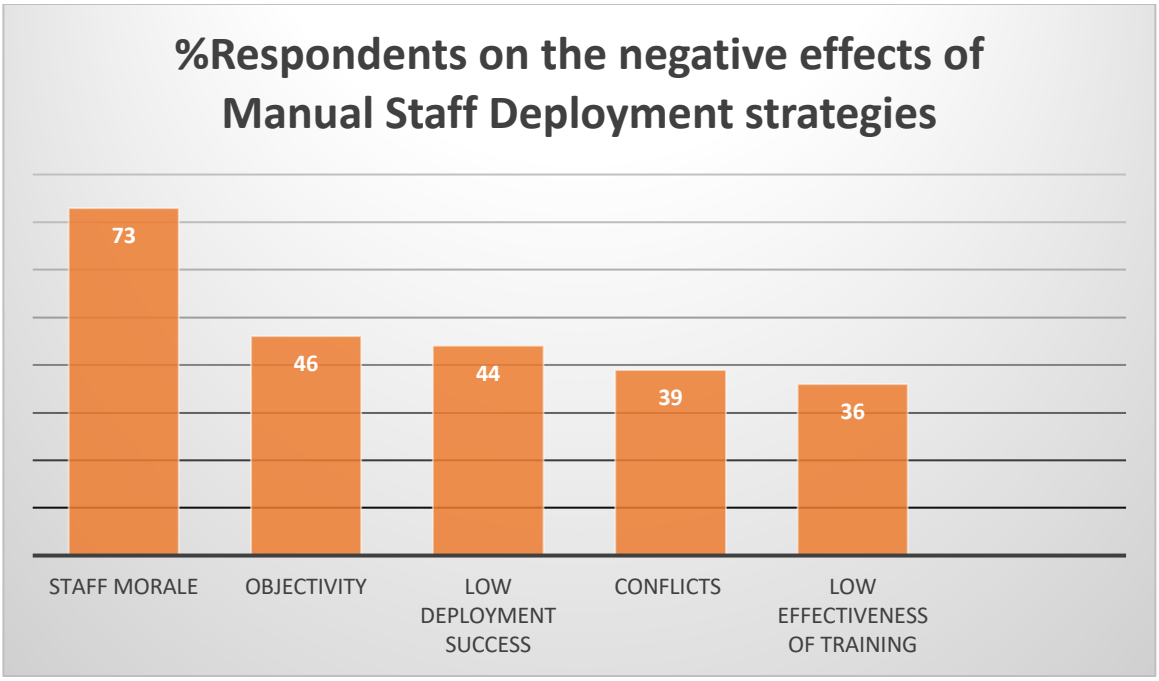
NGOs encounter significant challenges in efficiently deploying field staff due to their operations' intricate and ever-evolving nature, resulting in inefficiencies and escalating costs (Agnieszka Ziomek, 2020). Despite the potential benefits, only a handful of NGOs globally have utilized machine learning applications for staff deployment optimization. Examples include applications in crowdfunding, the International Federation of Red Cross and Red Crescent Societies, and Charity Catchfire (Mueller & Massaron, 2016).

The conventional manual methods employed for field staff deployment suffer from shortcomings in accuracy and timeliness, leading to inadequate resource allocation and diminished staff productivity. Utilizing erroneous or biased data can yield inaccurate predictions and undermine decision-making in machine-learning algorithms (Isnanto et al., 2020). The existing manual field staff deployment processes within NGOs are characterized by inefficiency, resulting in suboptimal resource utilization and diminished impact within target communities (Roberts & Downes, 2020). Manual deployment procedures consume considerable time and resources, consequently delaying responses to emergencies and unforeseen circumstances (Mccaffrey, 2020). Moreover, inadequate data quality and management practices contribute to inconsistencies, thereby compromising the accuracy of analysis and interpretation (Pan, 2020).

Application	*Manual Methods (Excel and calendar-based systems) e.g., data entry*	*Machine Learning Applications/Algorithms*
Comparative parameters		
Relative time consumed	400-500 minutes per 10000 fields (Johnson, 2020)	Can classify large volumes within seconds (Andročec & Vrček, 2018).
Rates of Error	1-2% per character and 10-20% per field (Data Entry Services)	0.5 to 5% per field (Towards Data Science)
Efficiency Rate	100-150 fields per hour	>1000 fields per hour
Cost Implications	$10-25 per hour per operator(data entry)	Low operational costs after set up (ScienceDirect)

NGOs frequently grapple with the challenge of reconciling the needs and preferences of field staff with the operational demands of the organization, resulting in potential conflicts and morale issues (Pan, 2021). The absence of accurate and current data concerning staff locations, experience, and training requirements poses challenges in making well-informed decisions regarding field staff deployment (Fuchs et al., 2021). Furthermore, the intricacy of machine learning algorithms often presents obstacles in interpreting and deploying algorithms (Manz, 2018).

Below is a bar graph indicating the extent to which the use of manual methods of staff deployment affects staff and NGO operations in the deployment field locations:



Inadequate data and information management systems in NGOs result in limited visibility into field staff operations, making it difficult to identify and address deployment challenges in real-time (Mccaffrey, 2020). Traditional statistical methods are insufficient for analyzing complex datasets and identifying patterns that can inform field staff deployment decisions. The lack of accurate and comprehensive data is a common problem in the field of data analytics (Kumar et al., 2022).

This research aims to address the challenges faced by non-governmental organizations (NGOs) in optimizing field staff deployment. The existing methods often lack efficiency and fail to consider multiple variables crucial for effective deployment, such as leave requests, staff experience, and deployment needs. By integrating machine learning clustering algorithms, particularly the k-means algorithm, this study seeks to develop a data-driven solution. The approach involves thorough data analysis, preprocessing, and algorithm evaluation, with a focus on user input through a web application (Agnieszka, 2020). The anticipated outcome is an improved, automated system providing NGOs with more advanced, accurate, and tailored field staff deployment recommendations.

Description	***Inefficiency***	***Source***
Current Staff Deployment Structure	Uneven distribution of staff	Bhargava & Snoap, 2003
Leave Management Process	Lack of optimization in handling time off	Kumar et al, 2022
Deployment Decision Making	Limited consideration of staff preferences	Isnanto et al., 2020
Staff Training Procedures	Inefficient methods for identifying needs	Dana et al., 2022
Previous Deployment Records	Lack of data utilization for future plans	Kibiwot, 2020
Manual Data Entry	Prone to errors, time-consuming	Bhargava & Snoap, 2003
Low Scalability	Inability to adapt to changing demands	Roberts & Downs, 2021
Manual Selection Decision-Making	Subject to biases, potential inefficiencies	Nyaga & Kimani, 2020
Inadequate Data Analytics	Limited insights from available data	Tyagi et al., 2023
Insufficient and low data quality	Misinformed decision/inaccurate prediction by data analytics algorithms	Pan, 2021
Time consuming processes	Delayed deployments during emergencies	Nyaga & Kimani, 2020

Employee preferences and needs not considered	Leads to low morale and conflicts at the workplace	Pan, 2021
Traditional statistical methods used	Limited ability to make sense of complex datasets and variables	Nyaga & Kimani, 2020

The existing NGO field staff deployment methods face significant inefficiencies that hinder optimal resource allocation. Manual decision-making processes introduce delays and errors, while limited consideration of variables such as leave requests, staff experience, and specific deployment needs contributes to suboptimal strategies. Scalability challenges arise as NGOs expand, impacting the management of larger datasets and diverse deployment scenarios (Agnieszka, 2020). Additionally, subjectivity in decision criteria and human bias can lead to inconsistencies. Overall, the absence of advanced analytics contributes to ineffective resource utilization, raising operational costs and decreasing efficiency in NGO operations.

Field staff deployment systems are software tools or platforms used to manage and optimize the deployment of field staff, such as technicians, sales representatives, or service support workers (Pan, 2021). These systems aim to improve the efficiency and effectiveness of field staff by facilitating scheduling, dispatching, tracking, monitoring, communication, and collaboration with field staff (Isnanto et al., 2020). They also include functionality for collecting and analyzing data on field staff activity.

Several field staff deployment systems are available on the market, offering a wide range of features and capabilities. Some common features include:

1. Scheduling and dispatch: These systems provide tools for scheduling and dispatching field staff to specific locations and tasks, ensuring efficient allocation of resources (Dana et al., 2022). Features may include real-time dispatch, routing optimization, and

mobile integration, enabling field staff to receive and respond to work orders on their mobile devices. By utilizing these tools, organizations can enhance the efficiency and effectiveness of field staff, reducing travel and administrative time and ensuring that field-based colleagues prioritize critical tasks ("Strategic human resource planning and staffing," 2019).

2. Tracking and monitoring: Field staff deployment systems often include tools for tracking and monitoring the location and activity of field staff, providing real-time visibility into their work. These tools may include features such as GPS tracking, location reporting, and activity logging, allowing organizations to track the movements and progress of field staff in real-time (Bisong, 2019). This capability is particularly valuable for organizations operating in large or dispersed geographical areas or those needing to monitor field staff working in remote or hazardous environments. In addition to enhancing efficiency and productivity, tracking and monitoring tools contribute to improved safety and compliance (Dana et al., 2022). They ensure that field teams are always aware of their surroundings and following proper protocols.

3. Communication and collaboration: Field staff deployment systems often include tools for enabling communication and collaboration between field staff, their supervisors, and other stakeholders, facilitating effective teamwork. These tools may encompass features such as messaging, document sharing, and task management, enabling assigned teams to stay connected and coordinate their work efficiently (Chen et al., 2022). By leveraging these tools, organizations can enhance communication and coordination between field staff and other stakeholders, mitigating the risk of errors or misunderstandings. This ensures that deployed staff have access to the most up-to-date information and resources, thus optimizing their performance.

4. Data collection and analysis: Field staff deployment systems frequently incorporate tools for collecting and analyzing data on field staff activity, such as task duration, work quality, and customer satisfaction. This data serves to optimize field staff deployment and enhance efficiency (Chen et al., 2022). These tools may comprise analytics dashboards, performance reports, and feedback forms, empowering organizations to understand and improve the performance of their field staff. By collecting and analyzing data on field staff activity, organizations can discern patterns and trends, enabling proactive adjustments to deployment strategies to improve efficiency. Consequently, organizations can reduce costs, enhance customer satisfaction, and drive sustained growth.

In addition to these common features, many field staff deployment systems also offer a range of specialized tools and capabilities to meet the specific needs and goals of different organizations. These may include features such as inventory management, customer relationship management, and field service management, among others ("impacts of HRIS implementation and deployment on HR professionals' competencies: An outline for a research program," 2008).

Overall, field staff deployment systems are a valuable tool for organizations looking to optimize the deployment and performance of their field staff (Mccaffrey, 2020). These systems can help organizations to improve efficiency and effectiveness, reduce costs, and enhance customer satisfaction, among other benefits.

However, it is essential that organizations carefully consider the specific needs and goals when selecting a field staff deployment system, as well as, diligently evaluate the features and capabilities of different systems to find the most applicable and efficient. In addition, regular

reviews and updates on these field staff deployment systems go a long way to ensure that they are meeting the changing needs of the organization and the field staff (Agnieszka Ziomek, 2020). This may involve training field staff on the use of the system, setting clear goals and metrics for performance, and continuously gathering and analyzing data to inform improvements and optimizations (Mccaffrey, 2020). By taking these steps, organizations can ensure that they are getting the most value and benefit from their field staff deployment systems and that their field staff will be well-equipped with the tools and resources they need to be successful in the field.

One of the major outcomes of such monitoring, evaluations, and reviews has been centered on the concept of staffing automation while ensuring fairness. The current field staff deployment systems have not adequately covered this aspect despite there being various machine-learning tools, techniques, and platforms (Agnieszka Ziomek, 2020). This paper seeks to highlight key steps to coming up with and developing a clustering-based machine learning algorithm for field deployment staffing in non-governmental organizations handling thousands of field teams.

The deployment of field staff has been a challenge in the industry, and to numerous Non-Governmental Organizations (NGOs) for several decades. These organizations rely on field staff to carry out their mandates, which range from humanitarian aid to community development, and disaster relief (Dana et al., 2022). The effective deployment of field staff is crucial to the success of these organizations in achieving their goals and objectives.

The traditional methods of deploying field staff are manual and involve a lot of effort and resources. The process typically includes collecting information on the staff's availability, their skills and experience, their training needs, and the location of deployment. This information is

then used to make deployment decisions. However, this process can be time-consuming and prone to errors, leading to ineffective deployment and the wastage of resources.

With the advancements in technology, the use of machine learning algorithms to automate field staff deployment has become a viable solution. The application of machine learning algorithms in the deployment process has the potential to improve the efficiency and accuracy of the deployment process (Bisong, 2019). In this research, the use of a machine learning clustering algorithm is proposed to automate field staff deployment for NGOs.

Currently, data analytics and machine learning algorithms are increasingly being leveraged by non-governmental organizations (NGOs) to enhance various aspects of their operations. In the realm of program management, data analytics play a crucial role in monitoring and evaluating project performance, enabling NGOs to track progress, identify trends, and assess the impact of their interventions. By analyzing large datasets, NGOs can gain valuable insights into beneficiary demographics, preferences, and needs, allowing for more targeted and effective program delivery (Andročec & Vrček, 2018). Machine learning algorithms are also being utilized to predict outcomes, such as the likelihood of project success or the risk of program failure, based on historical data and key performance indicators. These predictive analytics empower NGOs to make informed decisions, allocate resources strategically, and optimize program outcomes.

In addition to program management, data analytics and machine learning are transforming other areas of NGO operations, including fundraising, donor management, and advocacy. Advanced analytics techniques enable NGOs to segment donors effectively, personalize communications, and tailor fundraising strategies to maximize donor engagement and contribution. Machine learning algorithms can analyze donor behavior patterns, predict donor preferences, and optimize fundraising campaigns for better results(Mccaffrey, 2020). Moreover, in advocacy efforts, data analytics empower NGOs to identify key stakeholders,

assess public sentiment, and craft targeted advocacy campaigns that resonate with their audience. By harnessing the power of data analytics and machine learning, NGOs can enhance their effectiveness, efficiency, and impact in addressing pressing social and environmental challenges.

However, despite the significant potential benefits, notable gaps are emerging from the underutilization of data analytics and machine learning in NGO operations. One key gap is the limited capacity and expertise within NGOs to collect, analyze, and interpret data effectively. Many NGOs lack dedicated data analytics teams or resources to implement sophisticated analytics solutions, leading to underinvestment in data-driven decision-making processes. Additionally, data quality and accessibility remain significant challenges, with NGOs often struggling to access and integrate disparate datasets from various sources (Dana et al., 2022). This hampers their ability to derive meaningful insights and make informed decisions based on accurate and timely information. Furthermore, ethical considerations, such as data privacy and security concerns, pose additional barriers to the adoption of data analytics and machine learning in NGO operations. Addressing these gaps will require concerted efforts to build capacity, improve data infrastructure, and establish ethical guidelines to ensure responsible and effective use of data analytics and machine learning in NGO operations.

Gap in Use	**Challenges/Limitations**	**Proposed Solutions**
Limited capacity and expertise	Lack of dedicated data analytics teams and resources	- Provide training and capacity-building programs for NGO staff
	Insufficient funding for investing in data analytics capabilities	- Advocate for increased funding and support for data analytics initiatives within NGOs

	Difficulty in attracting and retaining data science talent	- Collaborate with academic institutions and offer internships or partnerships to access expertise
	Lack of awareness about the potential benefits of data analytics	- Conduct awareness campaigns and showcase success stories of data-driven decision-making in NGOs
Data quality and accessibility	Fragmented and siloed data sources	- Implement data integration solutions to centralize and standardize data
	Inadequate data governance frameworks	- Develop and enforce data governance policies and procedures to ensure data quality and security
	Limited access to relevant and timely data	- Establish partnerships with relevant stakeholders to access and share data resources
Ethical considerations	Data privacy and security concerns	- Develop and adhere to robust data privacy and security protocols to protect sensitive information
	Potential for bias in data collection and analysis	- Implement bias detection and mitigation techniques in data collection and analysis processes

	Transparency and accountability in algorithmic decision-making	- Document and communicate the decision-making process to ensure transparency and accountability
Infrastructure and technology	Outdated or inadequate IT infrastructure	- Invest in upgrading and modernizing IT infrastructure to support data analytics initiatives
	Lack of interoperability between systems	- Adopt standardized data formats and protocols to facilitate interoperability across systems
	Limited access to advanced analytics tools and platforms	- Explore open-source or cloud-based solutions that offer scalable and affordable analytics platforms

Moreover, available, and credible literature indicates the current challenges associated with the under usage of data analytics in the realm of NGO operations:

1. According to a report by the United Nations Development Programme (UNDP), approximately 1.3 billion people worldwide are served by NGOs, highlighting the significant scale of operations and the need for efficient resource allocation (UNDP, 2021).
2. A survey conducted by the Stanford Social Innovation Review found that 67% of NGOs cite resource constraints as a major barrier to achieving their mission, underscoring the importance of optimizing operational efficiency (SSIR, 2020).
3. The Global Humanitarian Assistance Report indicates that humanitarian organizations spent over $27.3 billion on staff costs in 2020, highlighting the substantial financial investment in field staff deployment (GHA, 2020).

4. In a study published in the Journal of International Humanitarian Action, researchers found that NGOs spend an average of 15-20% of their budget on staff-related expenses, further emphasizing the significance of efficient deployment strategies (JIHA, 2019).
5. Despite the increasing availability of data, a survey conducted by the World Bank revealed that only 30% of NGOs feel confident in their ability to use data effectively for decision-making, indicating a gap in data utilization capabilities (World Bank, 2021).

Clustering is a machine-learning technique that is used to group similar data points ("Probabilistic clustering," 2020). The objective of the clustering algorithm in this research is to group the field staff based on their availability, skills, and experience, as well as their location (Dana et al., 2022). The deployment needs of the organization was then be used to match the field staff with the deployment locations.

The study made use of the Python programming language and deployment on Heroku. The relevant Python packages and libraries were be applied in the algorithm, including Pandas, Numpy, and Scikit-learn. The dependent variables in the study included leave/time off requests, location (urban, rural, hardship), staff experience, staff training needs, and previous deployments. The independent variables included deployment needs, the number of staff needed, and staff/management preferences.

The methodology was carried out in several stages. The first stage involved the collection of data on the field staff and the deployment needs of the organization. The second stage involved the pre-processing of the data to remove any missing or irrelevant information (Dana et al., 2022). The third stage then incorporated the application of the machine learning

clustering algorithm to group the field staff based on their availability, skills, and experience, as well as their location.

The expected outcome of this research is the development of a machine learning clustering algorithm that can be used to automate the field staff deployment process for NGOs. The algorithm sought to improve the efficiency and accuracy of the deployment process, leading to better deployment decisions and the effective use of resources (Mccaffrey, 2020). The algorithm was also flexible, allowing for changes in the deployment needs of the organization and the availability of field staff.

In conclusion, the creation of a machine learning clustering algorithm for automating field staff deployment for NGOs has the potential to improve the efficiency and accuracy of the deployment process (Rosett & Hagerty, 2021). The study provides a new approach to addressing the challenge of deploying field staff in NGOs, offering a solution that has the potential to improve the effectiveness of these organizations in achieving their goals and objectives.

1.2.Problem Statement

Non-Governmental Organizations (NGOs) play a critical role in delivering essential services—especially in humanitarian aid, disaster relief, healthcare, and development. However, the continued reliance on manual and outdated deployment systems severely compromises their effectiveness. These systems often use spreadsheets, informal communication, and subjective decision-making to assign field staff, leading to significant operational disruptions.

The direct consequences of these inefficiencies are substantial:

- Delayed emergency response during disasters due to slow staff allocation.
- Underutilization of over 40% of skilled personnel in critical sectors such as healthcare (WHO, 2022).
- Wastage of approximately 20% of NGO project funds annually due to misallocation of staff and resources (KNBS, 2023).
- Inequitable distribution of qualified staff between urban and rural locations, exacerbating social disparities.

These issues result in slowed program execution, reduced community impact, donor dissatisfaction, and operational fatigue among field personnel. Yet, unlike corporate and governmental sectors that have embraced machine learning (ML) for predictive and efficient resource allocation, NGOs remain largely left behind.

This research aims to bridge this technological gap by developing a paired-algorithm clustering model—specifically combining Hierarchical Clustering and K-Means—to intelligently segment and allocate field staff based on factors such as experience, availability, location, and deployment history. By aligning key performance metrics with human-centric deployment needs, the model offers a data-driven solution that is scalable, efficient, and tailored to NGO operations.

This study not only proposes a novel machine learning approach but also presents a practical deployment framework that can be integrated into NGO staffing workflows, significantly reducing response time, minimizing resource wastage, and maximizing impact on the ground.

1.3.Main Objective

To create a paired-algorithm clustering machine learning algorithm to describe field staff deployment in NGOs and enhance efficient deployment processes.

1.4. Specific Objectives

The specific objectives of the study will be

- To conduct attribute analysis to identify 5 pertinent variables for clustering field staff in NGOs.

- To design and implement a paired-algorithm machine learning clustering model using Python, to cluster field staff for efficient deployment in NGOs
- To evaluate the algorithm's performance through the Silhouette Score Algorithm and Davies-Bouldin Index Algorithm.

1.5.Research Questions

- Which attributes can be used to cluster the field staff for deployment in NGOs?
- Which algorithm can use the identified attributes, to cluster the field staff deployment in NGOs?
- What is the performance of the developed algorithm?

1.6.Motivation for the Study

The motivation behind this project is to provide a predictive solution to the field staff deployment processes within non-governmental organizations (NGOs) through the application of machine learning algorithming. Traditional manual deployment methods are prone to inefficiencies, errors, and resource wastage, resulting in suboptimal resource allocation and diminished impact in target communities. By harnessing the capabilities of data analytics and machine learning, this project seeks to automate and optimize the field staff deployment process for NGOs.

Through the development of a sophisticated machine learning clustering algorithm, this project aims to uncover intricate patterns and groupings within field staff deployment data. By doing so, it enables NGOs to make data-driven decisions and allocate resources more effectively. This innovative approach has the potential to significantly enhance the efficiency, efficacy, and overall impact of NGOs by streamlining deployment procedures, cutting costs, and ensuring better alignment between staff competencies and deployment requirements. Moreover, by pushing the boundaries of research in data analytics and machine learning, this project contributes to the advancement of these fields while simultaneously addressing a critical and timely need within the NGO sector.

1.7. Significance of the Study

The development of a machine learning clustering algorithm for describing field staff deployment in NGOs holds significant implications for various stakeholders. Firstly, NGOs themselves stand to benefit from enhanced operational efficiency and resource optimization. By making use of predictive algorithms in the deployment process and considering factors such as staff experience, training needs, and location, NGOs can ensure better utilization of their workforce and improve response times, particularly in remote and challenging areas.

Secondly, the study contributes to the advancement of machine learning techniques for clustering and deployment optimization. By leveraging Python libraries and tools, the research demonstrates the efficacy of machine learning algorithms in addressing complex deployment challenges, thereby expanding the capabilities of these algorithms in practical settings.

Furthermore, the potential beneficiaries extend to the communities served by NGOs. Through improved deployment processes, NGOs can deliver aid and humanitarian services more effectively, reducing errors and ensuring better use of resources. This translates to quicker

and more successful project outcomes, ultimately benefiting the individuals and communities in need.

Lastly, the study contributes to the academic community's understanding of machine learning clustering algorithms' application in real-world scenarios, particularly within the context of NGO operations. The insights gained from this research can inform future studies and aid organizations in leveraging technology to enhance their operations, ultimately leading to greater impact and efficiency in humanitarian efforts.

CHAPTER TWO

2. LITERATURE REVIEW

2.2.Introduction

In this section, we embark on a comprehensive exploration of field staff deployment systems and their attributes, which serve as the foundation for optimizing the allocation of field staff resources (Andročec & Vrček, 2018). The theoretical review delves into the landscape of existing field staff deployment systems, dissecting their key features, advantages, and functionality. This insight into real-world deployment systems provides a practical context for the development of our clustering algorithm.

Workforce, Field Service Lightning, Jobber, Fleetio, Zoho Field Service, and HouseCall Pro are spotlighted in this review. Each system's distinct characteristics, such as scheduling, monitoring, and integration capabilities, are examined. Notably, these systems exhibit prowess in catering to a spectrum of organizational needs, from large enterprises to small and medium-sized businesses. The prominence of data analytics and reporting features in these systems underscores their commitment to performance enhancement.

Additionally, we scrutinize the attributes essential for clustering field staff for deployment. This section spotlights five pivotal variables that play a critical role in ensuring optimal staff allocation:

Leave/Time Off Requests in Field Staff Deployment Clustering: Acknowledging the significance of staff leave and time-off requests in clustering is paramount for maintaining operational efficiency and staff well-being (Andročec & Vrček, 2018). We emphasize the value of accommodating these requests and using clustering algorithms to manage absences proactively.

Previous/Current Deployment Location in Field Staff Deployment Clustering:Field staff's historical and present deployment locationsf are considered, recognizing the expertise and familiarity staff gain in specific areas. Clustering staff based on these attributes ensures that staff members with relevant experience are assigned to areas where their proficiency can be fully utilized.

Staff Experience: The experience of field staff is assessed based on factors like years in the organization and diverse tasks undertaken. Clustering individuals with similar experience levels aids in leveraging their collective expertise and enhance service delivery quality.

Deployment Needs: Addressing the specific requirements and demands associated with deployment assignments is essential. The urgency, scope of work, skills, and geographical location of deployments are considered within clustering algorithms to optimize staff allocation. Staff and Management Preferences: Individual staff preferences for location, schedule, and roles are acknowledged, ensuring job satisfaction and better performance. Management preferences are incorporated to align deployments with strategic objectives, enhancing organizational efficiency.

Furthermore, we explore the various clustering techniques available for optimizing field staff deployment. K-Means Clustering, Hierarchical Clustering, DBSCAN, Spectral Clustering, and Agglomerative Hierarchical Clustering are scrutinized. These techniques offer diverse methodologies for clustering field staff based on attributes and operational requirements. The theoretical review serves as a stepping stone for the empirical review, where we delve into the practical implementation and evaluation of clustering algorithms. We aim to

seamlessly bridge theoretical insights with real-world applications, ultimately enhancing the field staff deployment process for organizations.

3. Theoretical Review

4. A review of Previous and Existing Algorithms used by NGOs

Non-governmental organizations (NGOs) play a crucial role in delivering humanitarian aid and development assistance to communities worldwide. To optimize their operations and achieve greater impact, NGOs have increasingly turned to data analytics and machine learning techniques (Andročec & Vrček, 2018). Previous studies have explored various algorithms and approaches used in NGO operations, highlighting their strengths, limitations, and potential for improvement.

One prevalent algorithm applied in NGO operations is the use of geographic information systems (GIS) for spatial analysis and decision-making. GIS enables NGOs to map out areas of need, identify vulnerable populations, and plan resource allocation effectively. For instance, organizations like the International Federation of Red Cross and Red Crescent Societies (IFRC) utilize GIS to map disaster-prone areas and pre-position relief supplies, enhancing their preparedness and response capabilities (Anyango et al., 2017).

Another prominent algorithm is predictive analytics, which involves using historical data to forecast future events and trends. NGOs leverage predictive analytics to anticipate humanitarian crises, optimize resource allocation, and target interventions more efficiently (Bhargava & Snoap, 2003). For example, organizations like Mercy Corps use predictive analytics to predict food insecurity hotspots and plan food distribution programs accordingly.

Machine learning algorithms, including clustering algorithms like k-means, hierarchical clustering, and density-based clustering, are increasingly being employed in NGO operations to automate processes and improve decision-making. These algorithms analyze complex datasets to identify patterns, groupings, and relationships, enabling NGOs to optimize field staff deployment, prioritize interventions, and allocate resources effectively (Bhargava & Snoap, 2003). For instance, organizations like UNICEF use machine learning algorithms to analyze demographic data and identify at-risk populations for targeted interventions.

Despite the potential benefits, there are challenges associated with the application of these algorithms in NGO operations. These include issues related to data quality, accessibility, and privacy, as well as technical challenges in algorithm development, deployment, and interpretation (Bhargava & Snoap, 2003). Additionally, there is a need for greater collaboration and knowledge sharing among NGOs, academia, and technology partners to ensure the ethical and responsible use of data analytics and machine learning in humanitarian settings.

Clustering algorithms are fundamental tools in data analysis and machine learning, widely used in various domains, including non-governmental organizations (NGOs), to identify patterns and groupings in data. Several clustering algorithms have been developed, each with its unique characteristics, advantages, and limitations.

One of the most commonly used clustering algorithms is the k-means algorithm. K-means is a partitioning algorithm that aims to divide data into k distinct clusters based on similarity measures. It is widely used in NGO operations for tasks such as segmentation of beneficiary populations, resource allocation optimization, and field staff deployment. While k-means is relatively simple and efficient, its performance heavily depends on the initial selection of centroids and is sensitive to outliers.

Another popular clustering algorithm is hierarchical clustering, which organizes data into a hierarchical tree-like structure based on similarity measures. Hierarchical clustering is versatile and can handle various types of data, making it suitable for tasks such as community detection, network analysis, and program evaluation in NGOs **(**Bhargava & Snoap, 2003). However, hierarchical clustering can be computationally expensive, especially for large datasets, and may not scale well to high-dimensional data.

Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are also widely used in NGO operations. DBSCAN groups together data points that are closely packed, forming dense regions separated by sparser areas. This algorithm is particularly useful for identifying clusters of beneficiaries or regions with high need in humanitarian settings **(**Bhargava & Snoap, 2003). However, DBSCAN requires careful tuning of parameters and may struggle with datasets of varying densities or irregular shapes.

In recent years, machine learning techniques, including deep learning-based clustering algorithms, have gained popularity for their ability to handle complex data and extract meaningful representations automatically. Deep clustering methods, such as autoencoders and deep embedding clustering, have shown promise in tasks such as image classification, natural language processing, and anomaly detection in NGO operations (Anyango et al., 2017). These algorithms can learn hierarchical representations of data, capturing intricate patterns and relationships, but may require large amounts of labeled data and computational resources.

Accuracy levels of clustering algorithms vary depending on factors such as data quality, dimensionality, and algorithm parameters. While k-means and hierarchical clustering are

relatively simple and interpretable, they may struggle with datasets containing noise or overlapping clusters (Bisong, 2019). Density-based clustering algorithms like DBSCAN are robust to noise and outliers but may produce suboptimal results for datasets with varying densities or complex shapes. Deep clustering methods offer flexibility and scalability but require careful algorithm selection and tuning to achieve optimal performance.

5. Attributes considered when clustering Field Staff for deployment:

5.2.1. Leave/Time Off Requests in Field Staff Deployment Clustering

The efficient deployment of field staff in various organizations, particularly non-governmental organizations (NGOs), relies on a multitude of factors. Among these, leave and time-off requests play a significant role in determining the scheduling of field staff. Properly accommodating these requests is essential not only for the well-being of the staff but also for maintaining operational efficiency (Kibiwot, 2020).

Leave and time-off requests refer to the formal petitions made by field staff to be excused from work for a specific duration. These requests can stem from a variety of personal, health, or professional reasons. For instance, a staff member might request leave for a vacation, medical treatment, or attending training sessions.

In the context of clustering for field staff deployment, the consideration of leave and time-off requests is crucial. Staff members with approved leave requests should not be included in the clustering process for their absence duration, ensuring efficient deployment during their absence (Bisong, 2019). This prevents overburdening other staff members or disrupting the workflow during that period.

On the other hand, understanding historical leave patterns through data analysis can help in anticipating and accommodating future leave requests. By clustering staff members with similar leave histories, the scheduling process can be optimized to maintain operational efficiency (Anyango et al., 2017). For example, it might be beneficial to cluster staff members who consistently take leave during certain months or seasons, as this enables proactive planning for such absences.

Literature in the field acknowledges the importance of leave and time-off requests as a critical factor when clustering field staff. Advanced data analytics, machine learning, and clustering algorithms have proven effective in optimizing deployment to accommodate these requests while maintaining operational efficiency (Bisong, 2019). As such, this variable has been a fundamental component in modern field staff deployment strategies and the development of clustering algorithms.

In summary, the variable "leave/time off requests" plays a vital role in the clustering of field staff deployment, ensuring efficient utilization of resources and the well-being of staff members. Advanced analytics and clustering methods enable organizations to strike a balance between accommodating these requests and maintaining operational efficiency, making it a key consideration in modern field staff deployment algorithms.

5.2.2. **Previous/Current Deployment Location in Field Staff Deployment Clustering**

In the realm of field staff deployment for organizations, the consideration of previous or current deployment locations is an essential variable. Deploying staff efficiently and effectively involves taking into account the knowledge and experience of individuals in specific areas (Dana et al., 2022). This variable encompasses information about the urban, rural, or hardship areas in which staff members have previously worked or are currently deployed.

When it comes to clustering staff for deployment, understanding their previous or current deployment locations allows organizations to leverage the experience and familiarity that staff have with specific areas. For example, staff who have worked extensively in rural or remote locations may have developed specialized skills and knowledge that make them more suitable for deployments in similar areas (Fuchs et al., 2021). Clustering staff based on these experiences enables organizations to make optimal decisions in deploying the right staff to the right locations.

Additionally, analyzing this variable helps in ensuring the equitable distribution of field staff across various areas of operation. For instance, without clustering based on previous/current deployment locations, organizations might inadvertently concentrate staff in urban areas, neglecting rural or hardship areas, which are often underserved and may require more attention(Dana et al., 2022). The literature on field staff deployment acknowledges that considering previous/current deployment locations is a significant attribute when employing clustering algorithms.

By utilizing this variable, organizations can improve the effectiveness of their field staff deployment strategies. It not only enhances the efficiency of staff allocation but also contributes to achieving better outcomes in diverse operational contexts.

5.2.3. **Staff Experience**

The variable of "staff experience" is a pivotal element in the field staff deployment clustering process. It involves the consideration of the expertise and proficiency of field staff in their respective roles (Dana et al., 2022). This variable encompasses various aspects of an

individual's experience, such as the number of years in the organization, the diversity of tasks undertaken, and their skill development throughout their career.

When organizations incorporate staff experience into clustering algorithms, they gain a nuanced understanding of their workforce. For example, a staff member with several years of experience may have acquired a comprehensive knowledge of various aspects of their role, from problem-solving techniques to dealing with diverse challenges (Dana et al., 2022). Clustering individuals with similar experience levels enables organizations to capitalize on their collective expertise, improving the quality of service delivery.

Moreover, in contexts where specialized skills are required for certain tasks, staff experience plays a significant role. Consider a scenario where medical NGOs need to deploy healthcare workers to different regions(GHA, 2020). Those with experience in handling specific medical conditions or emergencies will be more effective in those situations. Clustering staff based on their experience ensures that the right individuals are assigned to areas that align with their proficiencies.

The literature on field staff deployment highlights the importance of staff experience as an essential attribute when creating clustering algorithms. By utilizing this variable, organizations can not only enhance the effectiveness of their deployments but also improve the overall quality of services provided (Dana et al., 2022). It is a crucial factor for optimizing decision-making processes in allocating staff resources.

5.2.4. **Deployment Needs**

The variable "deployment needs" is a significant variable to consider in field staff deployment clustering strategies. It encompasses the specific requirements and demands associated with

particular deployment assignments within an organization(JIHA, 2019). These needs can be diverse, including factors such as the urgency of deployment, the scope of work, the skills required, and the geographical location.

Deployments with high urgency, such as responding to a sudden disaster, demand immediate attention. In such cases, clustering algorithms should prioritize staff members who are readily available and capable of rapid response (Isanto et al., 2020). The variable "deployment needs" aids in categorizing these urgent situations and ensuring a swift and effective deployment response.

Another facet of deployment needs is the scope of work, which may vary widely across different assignments. For instance, a deployment to a remote rural area might require a different skill set than an urban assignment. Clustering staff based on these specific requirements ensures that the right individuals with the relevant skills are assigned to the corresponding tasks (Kibiwot, 2020).

Moreover, geographical factors play a significant role in determining deployment needs. Certain areas might pose security or logistical challenges(Bhargava & Snoap, 2003). Clustering algorithms consider these needs and assist in allocating staff members who are well-acquainted with the geographical nuances of the deployment location.

The data on field staff deployment recognizes the importance of the variable "deployment needs." It contributes to the effective allocation of resources and streamlines the deployment process by matching the unique requirements of each assignment with the skills and availability of staff members.

5.2.5. **Staff and Management Preferences**

Staff and management preferences play a key role in the field staff deployment clustering process. Understanding and incorporating these preferences into clustering algorithms can significantly enhance the efficiency of staff deployment while also improving job satisfaction and performance.

Staff members within an organization often have individual preferences for various aspects of their work. These preferences may include location preferences, schedule preferences, or specific roles they prefer to undertake (Dana et al., 2022). For instance, some staff members may have a strong preference for working in urban areas, while others may prefer rural or hardship postings. These individual preferences can be accommodated by clustering algorithms, ensuring that staff members are deployed in locations they find more suitable and are thus more likely to perform well.

Additionally, schedule preferences are a crucial consideration. Some staff may prefer working during specific times of the day, week, or year. By considering these preferences in the clustering process, organizations can improve employee satisfaction and motivation, resulting in better performance and reduced turnover (JIHA, 2019).

Management preferences, on the other hand, involve the deployment choices made by the organization's management based on their strategic goals and priorities. These preferences may include prioritizing specific types of deployments, areas of operation, or certain skills and qualifications in staff members (Dana et al., 2022). Clustering algorithms that incorporate management preferences can help optimize deployments in alignment with the organization's strategic objectives.

A plethora of research on staffing recognizes the significance of accommodating staff and management preferences within human resource management. By doing so, organizations can enhance their operational efficiency and employee satisfaction, ultimately leading to more effective field staff deployment and improved service delivery.

5.2.6. **Location of Deployments**

In the context of field staff deployment clustering, one of the critical variables to consider is the location of deployments. This variable pertains to the geographical areas where field staff members are assigned to perform their duties (Kumar et al., 2022). The location of deployments plays a pivotal role in ensuring that field staff are optimally distributed to meet the organization's objectives and serve the community effectively.

Organizations often have diverse needs that require field staff to be deployed in various locations. For example, a non-governmental organization (NGO) might need to provide healthcare services to both urban and rural communities. These differing deployment locations come with unique challenges and requirements(Kumar et al., 2022). In urban areas, staff may need to handle a higher volume of cases and work within better-established infrastructure, while rural or remote areas may present challenges related to accessibility, logistics, and resource constraints (Kibiwot, 2020).

Field staff deployment clustering algorithms need to take into account these varying deployment locations to optimize resource allocation. By categorizing deployment locations based on geographical characteristics and needs, clustering algorithms can help ensure that staff members with the most relevant skills and experience are assigned to each area (Kumar et al., 2022). For instance, staff with experience in urban healthcare settings can be clustered

and deployed to urban locations, while those with expertise in rural healthcare can be assigned to rural areas.

Moreover, the location of deployments may change over time due to shifting needs, emergencies, or community dynamics. Clustering algorithms should be adaptable and responsive to these changes, enabling organizations to quickly reallocate staff to address emerging situations.

Literature in the field of staff deployment clustering acknowledges that the location of deployments is a fundamental variable that influences the success of an organization's operations. By optimizing the distribution of field staff according to geographical factors, clustering algorithms can enhance service delivery, cost-effectiveness, and overall operational efficiency.

5.2.7. **Staff Training Needs in Field Staff Deployment Clustering**

Staff training needs constitute another vital variable in the realm of field staff deployment clustering. These needs encompass the specific skillsets, competencies, and qualifications required by field staff to effectively perform their duties in diverse deployment scenarios (Njiru, 2015). Understanding and accommodating these training needs is pivotal to enhancing the overall competence and performance of field staff.

Within the domain of field staff deployment, the requisite skillsets can vary significantly based on the tasks, roles, and deployment locations (Nyaga & Kimani, 2020). Certain deployments may necessitate specific technical expertise, such as medical or engineering skills, while others may demand soft skills like community engagement, language proficiency, or cultural sensitivity.

Accurately assessing and addressing staff training needs is integral to ensuring that field staff are adequately prepared for their assignments. Clustering algorithms employed in field staff deployment must consider the distinct training requirements associated with different locations and contexts (Pan, 2021). This entails the capacity to align field staff with the requisite training programs and resources, ensuring that they possess the competencies essential for their deployments.

Furthermore, the dynamic nature of staff training needs should not be underestimated. These needs can evolve over time due to changes in deployment requirements, advancements in technology, or shifts in the external environment. Effective clustering algorithms should be capable of recognizing these changes and adapting the training programs accordingly.

For instance, in the healthcare sector, the emergence of new medical procedures or equipment may necessitate ongoing training for field staff to maintain their proficiency (Pan, 2020). In the context of community development projects, shifts in best practices for community engagement may require updated training approaches.

The existing literature on field staff deployment clustering underscores the critical importance of staff training needs. It emphasizes that by integrating this variable into the clustering algorithms, organizations can ensure that their field staff are adequately equipped with the knowledge and skills essential for their roles (Kibiwot, 2020). This not only enhances the quality of service but also contributes to the safety and satisfaction of field staff.

5.3.Types of clustering techniques

- **K-Means Clustering:**

K-means is a centroid-based clustering technique that aims to partition data into 'k' clusters based on the mean of data points in each cluster. K-means is widely used in customer segmentation, image compression, and, in this project, optimizing field staff deployment by grouping staff into clusters based on attributes (Probabilistic Clustering, 2020). The technique minimizes the sum of squared distances between data points and their assigned cluster centers. It assigns each data point to the nearest cluster center, iteratively updating the centers until convergence.

Python code snippet:

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3)
kmeans.fit(data)
labels = kmeans.labels_
```

- **Hierarchical Clustering:**

Hierarchical clustering builds a tree-like structure of clusters, allowing for the identification of nested clusters at different levels. Hierarchical clustering is suitable for identifying hierarchical structures in data. It can help in understanding the optimal organization of field staff deployment. This type of clustering uses linkage methods like 'ward,' 'single,' or 'complete' to decide how to measure the distance between clusters (Probabilistic Clustering, 2020). The tree structure (dendrogram) is created by successively merging or splitting clusters.

Python code snippet:

```
from scipy.cluster.hierarchy import linkage, dendrogram
linkage_matrix = linkage(data, method='ward')
dendrogram(linkage_matrix)
```

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):**

DBSCAN is useful when there are irregularly shaped clusters, and noise is present in the data. It can be applied to field staff deployment to identify regions with varying staff densities. In

the theoretical background, DBSCAN forms clusters by connecting data points that are close to each other and have a sufficient number of neighboring data points within a specified distance (Strohmeier, 2022).

DBSCAN groups data points into clusters based on their density, with the ability to identify noise (outliers).

Python code snippet:

```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan.fit(data)
labels = dbscan.labels_
```

- **Spectral Clustering:**

Spectral clustering projects data into a lower-dimensional space using spectral decomposition and then applies standard clustering techniques in this transformed space. Spectral clustering is valuable when dealing with data that has non-linear relationships between variables. It can help optimize field staff deployment by capturing complex associations between staff attributes (Strohmeier, 2022). Spectral clustering employs graph theory concepts and spectral decomposition to transform the data into a more amenable space for clustering. It uses eigenvectors and eigenvalues to partition the data.

Python code snippet:

```
from sklearn.cluster import SpectralClustering
spectral = SpectralClustering(n_clusters=3)
spectral.fit(data)
labels = spectral.labels_
```

- **Agglomerative Hierarchical Clustering:**

Agglomerative hierarchical clustering starts with each data point as a single cluster and iteratively merges the closest clusters until a single cluster remains. This type of hierarchical

clustering is suitable for situations where you want to understand the hierarchy of clusters and how they merge. It can help in determining the optimal organizational structure of field staff.

Theoretical Background – Agglomerative clustering builds clusters using a bottom-up approach, merging data points or clusters with the smallest distance between them at each step (Strohmeier, 2022).

Python code snippet:

```
from sklearn.cluster import AgglomerativeClustering
agglomerative = AgglomerativeClustering(n_clusters=3)
agglomerative.fit(data)
labels = agglomerative.labels_
```

Paired combinational use of machine learning clustering algorithms involves utilizing multiple clustering techniques in tandem to leverage the strengths of each algorithm and overcome their individual limitations (Strohmeier, 2022). This approach aims to enhance the overall performance and robustness of clustering solutions by integrating complementary methods and mitigating the weaknesses inherent in any single algorithm. In this detailed review, we will explore several paired combinations of clustering algorithms, evaluate their effectiveness in various scenarios, and ultimately suggest the best combination based on empirical evidence and theoretical considerations.

5.3.1. K-means and DBSCAN:

- K-means is a centroid-based algorithm that partitions data into k clusters based on similarity measures.
- DBSCAN is a density-based algorithm that identifies clusters as dense regions separated by sparser areas.

- Paired combination: K-means is initially used to partition the data into clusters, followed by DBSCAN to refine the clusters by identifying noise points and adjusting cluster boundaries based on local density information (Strohmeier, 2022).

- Justification: This combination leverages the efficiency of K-means in partitioning data while benefiting from DBSCAN's ability to handle noise and identify arbitrary-shaped clusters.

5.3.2. Hierarchical clustering and Gaussian Mixture Algorithms (GMM):

- Hierarchical clustering organizes data into a tree-like structure based on similarity measures.

- GMM algorithms data points as samples drawn from a mixture of several Gaussian distributions.

- Paired combination: Hierarchical clustering is first used to obtain an initial clustering hierarchy, which is then fed into GMM to algorithm the clusters as Gaussian distributions and estimate parameters such as means and covariances (Strohmeier, 2022).

- Justification: This combination combines the flexibility of hierarchical clustering with the probabilistic algorithming capabilities of GMM, allowing for the identification of complex cluster structures and the estimation of cluster uncertainty.

5.3.3. Spectral clustering and Affinity Propagation:

- Spectral clustering projects data points into a low-dimensional space using spectral techniques before performing clustering.

- Affinity Propagation identifies exemplar points that best represent clusters and assigns each data point to the nearest exemplar (Strohmeier, 2022).

- Paired combination: Spectral clustering is applied to capture the global structure of the data, followed by Affinity Propagation to refine the clusters based on local similarities and exemplar selection.

- Justification: This combination combines the spectral embedding capabilities of spectral clustering with the exemplar-based clustering approach of Affinity Propagation, allowing for the identification of both global and local structures in the data.

5.3.4. Density-based clustering (DBSCAN) and Agglomerative clustering:

DBSCAN identifies clusters as dense regions separated by sparser areas. Agglomerative clustering is a hierarchical clustering algorithm that starts with each data point as a singleton cluster and iteratively merges clusters based on distance (Strohmeier, 2022).

Paired combination: DBSCAN is first applied to identify dense regions and core points, followed by Agglomerative clustering to hierarchically merge the identified clusters.

Justification: This combination leverages the noise-handling capabilities of DBSCAN and the hierarchical merging strategy of Agglomerative clustering to produce robust and interpretable cluster hierarchies.

From the foregoing, the paired combinational use of K-means and DBSCAN emerges as one of the most desirable pairs for a wide range of scenarios. K-means provides an efficient initial partitioning of the data, while DBSCAN offers robust noise handling and the ability to identify arbitrary-shaped clusters (Strohmeier, 2022). This combination strikes a balance between efficiency and effectiveness, making it suitable for diverse clustering tasks in NGO operations. However, hierarchical and k-means algorithm pairing stands out for the below reasons.

1. **Hierarchical Structure Exploration:** Hierarchical clustering allows for the exploration of the hierarchical structure of the data, providing insights into potential cluster hierarchies at different levels of granularity. This hierarchical view of the data

can be valuable for understanding the natural grouping tendencies within the data, which may not be captured by k-means or DBSCAN alone (Pan, 2021).

2. **Robustness to Noise and Irregular Shapes:** Hierarchical clustering is robust to noise and can handle irregularly shaped clusters effectively. This is particularly advantageous in scenarios where the data contains noise or where the clusters exhibit complex shapes that may not be well-suited for partitioning-based algorithms like k-means. In contrast, DBSCAN is also capable of handling noise and irregular shapes but may require careful parameter tuning, which can be challenging in practice (Mccaffrey, 2020).
3. **Flexible Number of Clusters:** Hierarchical clustering does not require specifying the number of clusters a priori, allowing the algorithm to discover the natural clustering structure of the data. This flexibility is beneficial in scenarios where the optimal number of clusters is unknown or where the data contains varying degrees of cluster density. In contrast, k-means requires the user to specify the number of clusters, which may not always be straightforward to determine, and DBSCAN requires setting parameters that control cluster density, which can be sensitive to the choice of parameters (Mccaffrey, 2020).
4. **Initial Cluster Assignment:** Hierarchical clustering can be used to generate initial cluster assignments that can serve as input to k-means clustering. This initialization step can help improve the efficiency and effectiveness of k-means by providing an initial partitioning of the data based on the hierarchical clustering results (Pan, 2020). In contrast, using DBSCAN as an initialization method may not be as straightforward, as DBSCAN relies on density-based criteria for cluster formation, which may not align with the objectives of k-means clustering.

Overall, the combination of hierarchical clustering and k-means clustering offers a versatile and effective approach to clustering analysis, particularly in scenarios where the data contains

complex structures, varying densities, and unknown or flexible numbers of clusters. This approach leverages the strengths of both algorithms to produce robust and interpretable clustering results.

6. Empirical Review

This article by Anyango et al., 2017 discusses the challenges and limitations of manual data entry in various sectors in Kenya, such as organizations, agriculture, financial institutions, and healthcare. They compare manual data entry with machine learning approaches, highlighting the advantages of machine learning in terms of efficiency, accuracy, and data quality.

The study by Anyango, Ngumi, and Owino (2017) titled "Manual Data Entry versus Machine Learning: A Comparative Analysis of Efficiency and Accuracy in Kenyan Healthcare Organizations" aimed to compare the efficiency and accuracy of manual data entry with machine learning in Kenyan healthcare organizations. The key findings, gaps, and recommendations from the study are summarized below:

Efficiency: The study found that machine learning significantly improved data entry efficiency compared to manual data entry. Machine learning algorithms could process and enter data faster, reducing the time required for data entry tasks.

Accuracy: Machine learning also demonstrated higher accuracy in data entry compared to manual methods. It reduced errors and inconsistencies that are common in manual data entry, leading to more reliable and trustworthy data.

Human Errors: The study highlighted that manual data entry was prone to human errors due to factors like fatigue, distractions, and variations in data interpretation. Machine learning algorithms, on the other hand, showed consistent performance without being affected by human-related issues.

Implementation of Machine Learning: Based on the findings, the study recommended that healthcare organizations consider integrating machine learning into their data entry processes. This can lead to improved efficiency, accuracy, and data quality.

Training and Capacity Building: To fully harness the benefits of machine learning, the study suggested providing training and capacity-building initiatives to healthcare staff. This would help them understand and use machine learning tools effectively in their daily tasks.

Further Research: The study encouraged further research in the field of machine learning applications in healthcare organizations. Additional studies could explore the impact of machine learning on other healthcare operations, such as patient diagnosis, treatment recommendations, and resource allocation.

The study revealed the advantages of employing machine learning over manual data entry in Kenyan healthcare organizations. It highlighted the potential of machine learning to enhance efficiency and accuracy while reducing human-related errors in data entry tasks. However, the study also pointed out the need for more extensive research and capacity-building efforts to fully embrace and implement machine learning technologies in healthcare settings.

Nyaga and Kimani (2020) conducted a comparative analysis of manual data entry and machine learning for data processing in the Kenyan agricultural sector. They found that manual data entry resulted in human errors, data quality issues, and delays in decision-making

processes. They concluded that implementing machine learning in data processing and analysis would reduce these challenges and improve overall organizational performance.

The researchers recommended the adoption of digital systems for data management to improve data accuracy, reduce manual errors, and enhance operational efficiency. They also suggested the use of digital solutions to automate and simplify data entry tasks, reduce human errors, and enhance overall data quality. Additionally, they proposed the use of digital systems to facilitate scalability and handle larger volumes of data.

6.2.1. A Machine learning algorithm for task allocation (Kibiwot, 2020)

The research presents a sophisticated predictive algorithm designed to optimize task allocation within an organizational context. This algorithm is composed of several key components, each playing a crucial role in the overall functionality and effectiveness of the algorithm.

The initial phase of the algorithm involves the extraction, cleaning, and merging of data. This process draws data from various Human Resources (HR) and Real-Time (RT) sources, ensuring that the information is comprehensive and up-to-date. The data cleaning process removes any inconsistencies or inaccuracies, while the merging process integrates data from different sources into a cohesive dataset.

These attributes provide a robust foundation for analyzing and predicting the most suitable task allocations for employees. The system leverages a command-line interface to facilitate user interaction and data processing. This interface is designed to be user-friendly and efficient, allowing for seamless navigation and operation. The primary functionalities of the system include:

- Displaying task categories: The system categorizes tasks based on predefined criteria, enabling users to understand the nature and requirements of each task.

- Identifying eligible employees: By analyzing the data, the system identifies employees who are eligible for specific tasks, taking into account their attributes and historical performance.
- Employee selection preferences: The system also allows for the customization of employee selection preferences, giving users the flexibility to prioritize certain attributes or criteria in the task allocation process.

Overall, the predictive algorithm integrates data extraction, cleaning, merging, and user-friendly system components to create a comprehensive and effective algorithm for task allocation. This approach not only enhances the efficiency of task assignment but also ensures that tasks are allocated to the most suitable employees, ultimately improving organizational productivity and employee satisfaction.

The shortcoming of this algorithm is that, despite the developed technique, there are additional manual steps involving employees choosing teammates manually and supervisors choosing or approving the same.

6.2.2. Improving recruit distribution decisions in the US Marine Corps (Bhargava & Snoap, 2003)

The Recruit Distribution Decision Support System (RDdss) is designed to improve recruit distribution decisions in the US Marine Corps. The RDdss is introduced as a system to enhance recruit distribution decisions, addressing existing challenges and streamlining the process. It explains the process of assigning recruits to entry-level schools, emphasizing its significance and the challenges it presents, such as ensuring recruits are assigned to the appropriate training programs that match their skills and the Corps' needs.

The system highlights limitations and issues with the current recruit distribution algorithming (RDM) system, underscoring the need for a more efficient and effective approach.

The recruit distribution is described as a multi-objective problem, identifying areas for improvement within the current system. RDdss aims to enhance the problem-solving process and information systems related to recruit distribution, ensuring a more streamlined and accurate distribution process.

Shortfall penalties are introduced to differentiate between schools, helping prioritize the distribution process based on the needs and priorities of various training schools. The primary objectives of the RDdss are to place the right Marine in the right place at the right time, optimizing the distribution process to meet these goals effectively.

The system preprocesses data to allow decision-makers to adjust weights for multiple distributions, facilitating more informed and effective decision-making. It highlights the limitations of using the utility score for comparative assessments, emphasizing the need for more reliable metrics. Metrics such as Fill, Wait, Unassignables, and Fit are described to evaluate the effectiveness of the distribution solutions.

The RDdss explains the use of trade-off parameters and comparing scores to make distribution decisions, helping decision-makers choose the best solutions based on various criteria. An interactive procedure for finding Pareto optimal solutions is introduced, aiding in the identification of the best possible solutions considering multiple objectives. The system includes features like graphical output and graphs to examine the quality of the solutions, providing a visual representation of the distribution outcomes.

Key components of the RDdss include the Switchboard, Relational database, Preprocessor, Assignment algorithm, Solver, and Analyzer. These components work together to facilitate the recruit distribution process. The system explains how data tables are extracted and preprocessed for the assignment algorithm, ensuring that the data used is accurate and ready for analysis.

The steps involved in setting up and executing the RDdss are detailed, providing a guide for users on how to effectively utilize the system. The potential benefits of the RDdss for the US Marine Corps are summarized, including the expected improvements in recruit distribution efficiency and effectiveness. Organizational challenges that may need to be addressed for successful implementation are also mentioned. Finally, the mathematical algorithm used for solving the recruit distribution problem is described, providing a technical foundation for the RDdss.

These components collectively form the RDdss algorithm, which encompasses data processing, optimization, decision-making, and system implementation to enhance recruit distribution decisions within the US Marine Corps. The identified shortfall here is that the RDss is based on a mathematical algorithm that renders is unscalable. The fact that it is not based on machine-learning techniques also informs to the possible manual nature of data preprocessing, storage and algorithming.

6.2.3. Clustering and Visualizing the Status of Child Health in Kenya: A Data Mining Approach (Njiru, 2015)

The research titled "Clustering and Visualizing the Status of Child Health in Kenya: A Data Mining Approach" outlines a comprehensive data mining and analytics algorithm that includes several critical steps. First, the introduction and problem statement acknowledge the necessity of clustering regions in Kenya that share similar characteristics impacting child health. This recognition forms the foundation of the study's objectives.

Next, the methodology section details the use of an explanatory research design, which combines exploratory research, descriptive statistics, data mining, and clustering techniques. The study adopts the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, providing a structured approach to data mining.

In the data preprocessing stage, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the high-dimensional input data. This step is crucial for making the data more manageable and ensuring the analysis is efficient and effective.

Visualization and exploration of the data follow, utilizing various visual tools such as scree plots, 2D/3D projections, scatter plot matrices, and biplots. These visualizations help explore and identify patterns within the child health data, offering a clearer understanding of the underlying trends and relationships.

The core of the analysis is the cluster analysis, which employs K-means clustering to categorize counties into three distinct clusters: well-off, most marginalized, and moderately marginalized. This categorization helps in understanding the distribution of child health status across different regions.

A comparison of clusters is then conducted, examining various child health indicators, including literacy rates, healthcare delivery efficiency, and fertility rates. This comparison provides a detailed understanding of the differences and similarities among the clusters.

The results and insights section offers valuable insights into the characteristics of counties within each cluster. It identifies counties with particularly high or low child health indicators, providing a nuanced understanding of the regional disparities.

Finally, the study highlights the contribution and importance of its findings. It emphasizes the significance of identifying the status of child health in Kenya and underscores the potential of the study to guide improvements in health policies and interventions.

However, the study also notes several shortfalls. These include limitations in data availability and quality, potential biases in data collection, and the challenges in generalizing the findings to other contexts without further validation. Additionally, while PCA helps

manage data dimensionality, it may lead to the loss of some nuanced information that could be critical for specific local contexts. These shortfalls point to areas for further research and improvement in future studies.

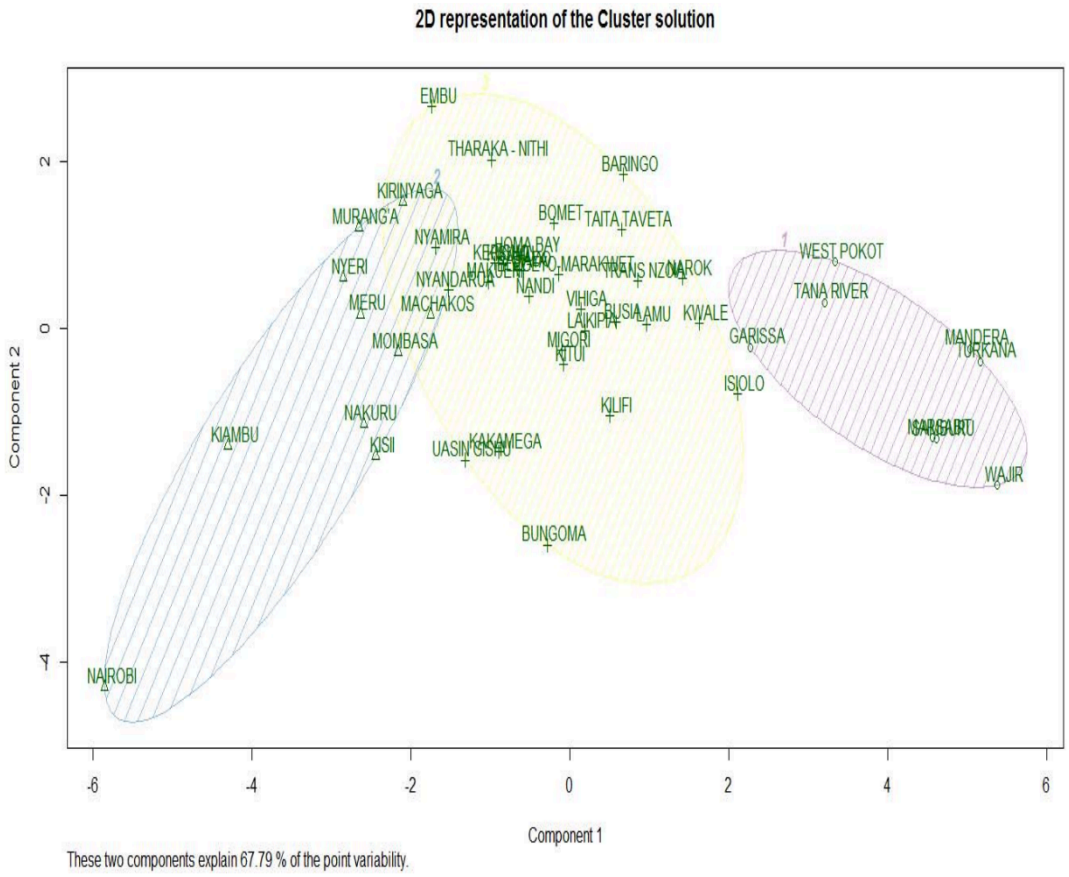


Figure 29-K-Means clustering results

Picture 1- K-Means clustering results (Njiru, 2015).

This data mining and analytics algorithm effectively processes and analyzes child health data to categorize counties into distinct clusters, providing valuable insights for policymakers and healthcare practitioners in Kenya. The shortfall identified with the algorithm developed here is that it is based on a relatively small amount of data, is less scalable or sophisticated. It is also noteworthy that the algorithm makes use of just one algorithm.

6.2.4. A Summary of Relevant Works in ML-Based Deployment

Study	Sector	Approach	Key Insight	Limitation
Kibiwot (2020)	HR/Task Allocation	Predictive ML	Dynamic task assignment based on staff profiles	Still partially manual and non-scalable
Njiru (2015)	Public Health	K-Means	Regional health clustering in Kenya	Used only one algorithm; lacks deployment logic
Bhargava & Snoap (2003)	Military	Rule-based + Optimization	Multi-objective recruit distribution	Not ML-based; not scalable for NGO use
Anyango et al. (2017)	Healthcare	Manual vs ML Data Entry	ML improves speed & accuracy	Doesn't address deployment/clustering

7. Limitations and Challenges in the Application of Existing Deployment Systems

There are several existing and potential limitations that NGOs encounter when using field staff deployment systems. Some of the most common limitations include:

Cost: Field staff deployment systems can be expensive to purchase and implement, especially for NGOs that may have limited budgets (Mccaffrey, 2020). NGOs may need to carefully consider the cost of a field staff deployment system and determine whether it is a worthwhile investment given their specific needs and goals.

Training and implementation: Field staff deployment systems can be complex to use and may require a significant amount of training and support to get up and running(Mccaffrey,

2020). This can be especially challenging for NGOs that may not have a lot of experience with these types of systems and may require significant investment in training and support to ensure that field staff can use the system effectively.

Integration with other systems: Field staff deployment systems may not always integrate seamlessly with other systems and tools that NGOs use, such as CRM or ERP systems (Isnanto et al., 2020). This can make it difficult for NGOs to get the full benefit of the system and may require significant effort to set up and maintain the integration.

Data privacy and security: Field staff deployment systems often involve the collection and processing of sensitive data, such as personal information and location data **(**Bhargava & Snoap, 2003). This can raise concerns about data privacy and security, and NGOs may need to carefully consider these issues and take steps to protect the data that they collect and process.

Complexity: Some field staff deployment systems can be quite complex, with a wide range of features and capabilities. This can make them difficult for NGOs to understand and use effectively, especially if they are not familiar with these types of systems. NGOs may need to carefully consider the complexity of a field staff deployment system and determine whether it is a good fit for their needs and capabilities (Isnanto et al., 2020).

Limited customization: Some field staff deployment systems may offer limited customization options, which can make it difficult for NGOs to tailor the system to their specific needs and goals (Isnanto et al., 2020). This can be a significant limitation, especially if an NGO has unique requirements or processes that are not well-supported by the system.

Limited scalability: Some field staff deployment systems may not be able to handle large volumes of data or may not be able to scale to meet the needs of an organization as it grows (Chen et al., 2022). This can be a significant limitation for NGOs that are expanding their operations or that need to handle large volumes of data.

Limited integration with other tools: Some field staff deployment systems may not offer integration with other tools and systems that NGOs use, such as accounting software or project management tools. This can make it difficult for NGOs to get the full benefit of the system and may require significant effort to set up and maintain the integration.

Limited support: Some field staff deployment systems may not offer extensive support options, which can be a significant limitation for NGOs that need help with training, implementation, or troubleshooting. This can be especially challenging for NGOs that may not have a lot of experience with these types of systems.

Limited reporting and analysis: Some field staff deployment systems may not offer extensive reporting and analysis capabilities, which can make it difficult for NGOs to understand and optimize the performance of their field staff (Fuchs et al., 2021). This can be a significant limitation, especially if an NGO needs to track and analyze data to make informed decisions about deployment and performance.

8. Gap in Literature

- Most ML deployment models focus on corporate or government contexts.
- NGOs are underrepresented in predictive staffing models.
- Existing studies don’t combine multiple clustering algorithms.

- Few studies assess deployment attributes like leave, hardship locations, or staff preferences.
- Lack of real-time or semi-automated decision systems tailored to NGO realities.

9. Contributions of this Study

- Introduces a novel, paired-algorithm clustering framework customized for NGO field deployment.
- Incorporates human-centric deployment factors like staff preferences and past deployment records.
- Evaluates and compares performance using industry-accepted metrics (Silhouette Score, DBI).
- Provides scalable, adaptable ML models deployable via cloud platforms (e.g., Heroku).

10. Conceptual Framework

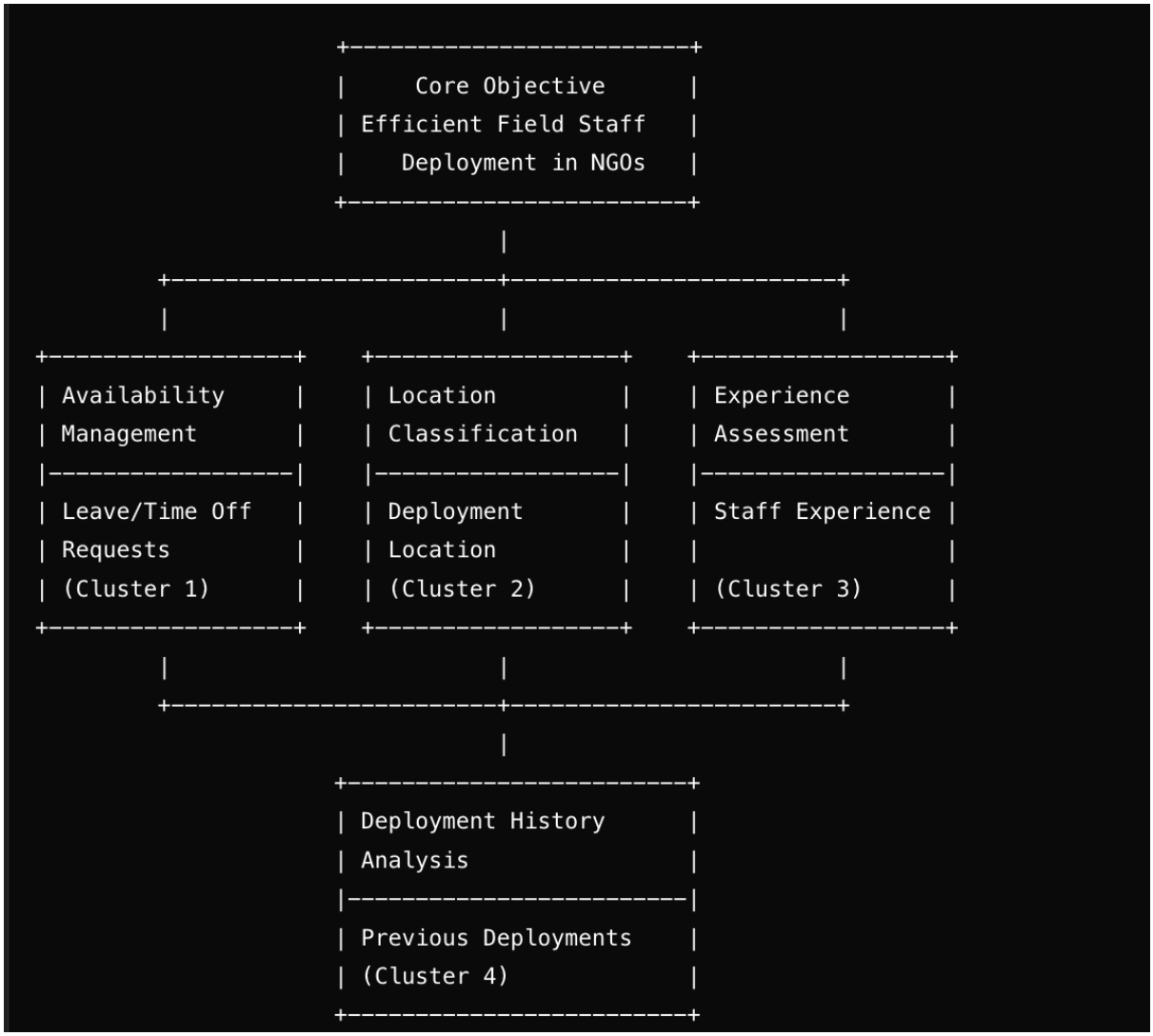


Figure 2: *Conceptual Framework*

10.2. Operationalization of Variables

Variable	Indicator	Data to be collected

Leave/Time Off Requests	• **Available (Categorical)** • **Unavailable (Categorical)**	• **Leave Request Date (Data Type: Date)** • **Duration of Leave (Data Type: Numeric - Days)** • **Staff Name (Data Type: Text)**
Previous/Current Deployment Location	• **Previous Deployment Location (Categorical)** • **Current Deployment Location (Categorical)** • **Duration in Current Location (Days) (Integer)** • **Rural, City, Town, Hardship (Categorical)**	• **Deployment Location (Data Type: Text)** • **Dates (Data Type: Date)** • **Duration (Data Type: Text)**
Staff Experience	• **Trained (Categorical)** • **Experienced (Categorical)**	• **Number of Years of Experience (Data Type: Numeric - Years)**

	• **New (Categorical)**	• **Training Need (Data Type: Text)**

Table 3: *Operationalization of Variables*

CHAPTER THREE

3. METHODOLOGY

3.1. Mixed-Methods Approach

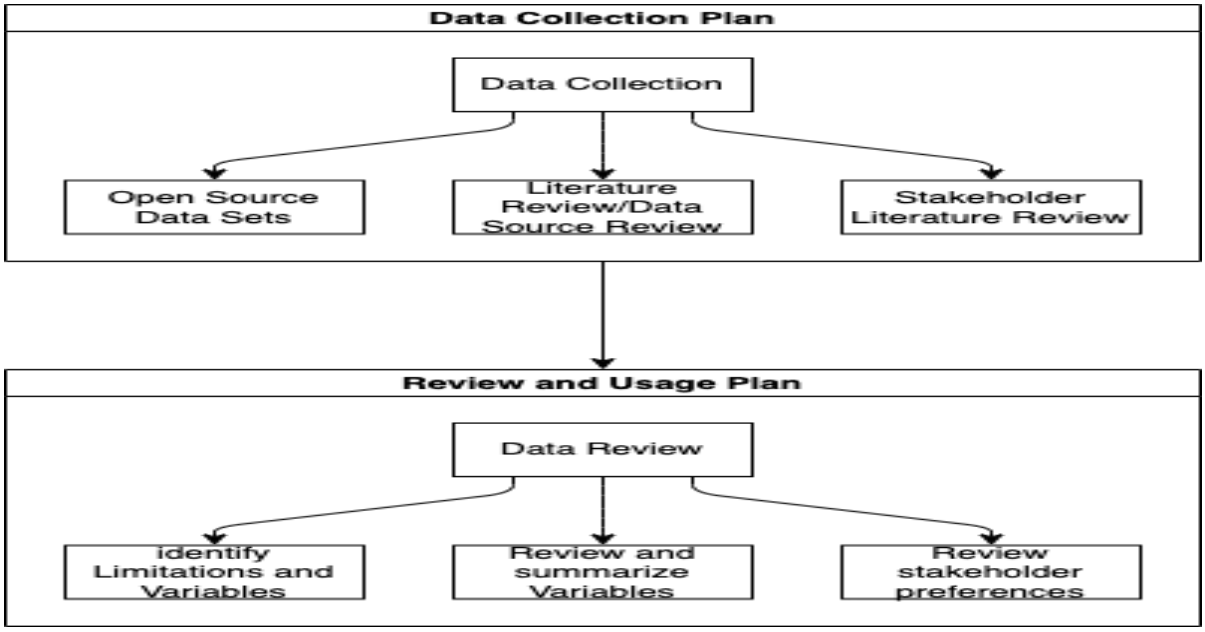
This study adopted a mixed-methods research design that combines quantitative clustering analysis with qualitative reasoning drawn from NGO operational practices. This integration enabled a robust approach to modeling field staff deployment by grounding machine learning outputs in practical NGO staffing dynamics. The quantitative analysis informed the development and evaluation of clustering models, while qualitative interpretation guided the choice of deployment attributes and algorithm assessment. Quantitative ML outputs were interpreted through the lens of NGO field operations. Qualitative reasoning was applied to select features, define cluster utility, and evaluate deployment logic.The combined method ensured technical rigor and real-world applicability.

Subsequently, the study research focused on a specific application of machine learning clustering algorithms for field staff deployment, which is the optimization of field team composition and scheduling (Kumar et al., 2022). This involved the use of the k-means and agglomerative algorithm as a clustering algorithm to group similar variables tied to specific employees based on their attributes such as leave or time off requests, training needs, previous deployments, job experience, other unique preferences, and historical data of field staff visits.

In the implementation of the algorithm, the project made use of Python for data preprocessing, cleaning, and analysis, as well as for implementing the k-means algorithm ("Probabilistic clustering," 2020). For visualization, the research and documentation focused on the application of libraries such as matplotlib and Seaborn to create maps and plots to illustrate the clustering results.

3.2. Data Collection and Variables

Literature Review and Data Source Review



This study focused on developing a machine learning clustering algorithm for NGO field staff deployment. Dependent variables included leave/time off requests, deployment location (urban, rural, hardship), staff experience, training needs, and past deployments. Independent variables encompassed deployment needs, staff quantity required, and staff/management preferences.

Data collection involved reviewing existing NGO staff deployment systems and identifying variables crucial for optimal deployment (Pan, 2021). Machine learning clustering, implemented using Python and libraries like sci-kit-learn, pandas, and NumPy, forms the core of the algorithm (Strohmeier, 2022). Evaluation employs metrics like accuracy, precision, recall, and sensitivity analysis.

By considering important variables such as leave/time off requests, location, staff experience, staff training needs, previous deployments, deployment needs, the number of staff needed, staff/management preferences, and using machine learning techniques to analyze the

data, the aim was to create a algorithm that is accurate and generalizable to a wide range of NGOs (Pan, 2021). In the clustering machine learning algorithm for automating field staff deployment for NGOs, the variables that were identified as dependent and independent variables played different roles in the algorithm.

The dependent variables, such as leave/time off requests, location (urban, rural, hardship), staff experience, staff training needs, and previous deployments, were used as the determinant variables of the algorithm. These variables were used to create a profile for each field staff member, which were then used to group similar staff members together (Strohmeier, 2022). Clustering algorithms, such as k-means and hierarchical clustering, were used to group the staff members based on their profiles.

The independent variables, such as deployment needs, number of staff needed, and staff/management preferences, were used to inform the algorithm's output by querying the profiles of the staff members. These variables were applied to determine the optimal deployment plan for the staff members in each cluster (Mccaffrey, 2020). For example, if there is a high need for staff in a certain location, the algorithm prioritized deploying staff members who are located close to that location or have experience working in that area. Similarly, if a certain staff member has requested time off, the algorithm also considered that when determining their deployment schedule.

The algorithm included the combination of these variables in order to determine the optimal field staff deployment. The algorithm went ahead to take into account the staff members' profiles, which are determined by the dependent variables, and the deployment needs, which are determined by the independent variables, to find the best match between the staff and the deployment locations (Pan, 2021).

The clustering machine learning algorithm then used these variables to predict the best deployment for the staff members. The algorithm learnt from the data provided and will use these predictions to optimize the field staff deployment process (Pan, 2021). By taking into account the various factors that are important in determining the optimal field staff deployment, the algorithm was able to make more accurate and efficient deployment decisions.

Effective data collection is crucial for building a successful machine learning clustering algorithm to optimize field staff deployment. This section outlines the steps taken to collect relevant data for the research.

- **Literature Review and Data Source Review**:
    - **Gathering Data from Open-Source Repositories**: The research began by sourcing data from an open-access repository – Kaggle. The platform provided access to a wealth of datasets related to field staff deployment, including historical deployment records, staff attributes, and geographic information.
    - **Reviewing Existing Systems and Algorithms**: In addition to open-source datasets, the research conducted a comprehensive review of existing field staff deployment systems and machine learning clustering algorithms. This review helped identify state-of-the-art practices and potential limitations in current approaches.
- **Identification of Relevant Variables**:
    - **Table summarizing Variables**: To streamline the research process, a table was created to summarize the key variables used in the analysis. These variables include attributes related to field staff, deployment needs, geographic locations, and other relevant factors.

Table 4: *Relevant Variables:*

Variable Name	Description
Staff Experience	Years of experience of field staff
Leave/Time Off Req.	Requests for leave or time off
Training Needs	Staff training requirements
Deployment Location	Rural, urban, or hardship area
Previous Deployments	Historical records of staff deployments
Deployment Needs	Required number of staff for specific tasks
Staff Preferences	Individual preferences and availability

- **Review of Literature Sources**:
    - **Table summarizing Data Sources**: A separate table summarizes the literature sources used in the research. This includes publications, reports, and academic papers that provide insights into field staff deployment and related datasets.

Source Title	Author(s)	Publication Year
Clustering and Visualizing the Status of Child Health in Kenya: A Data Mining Approach.	Nicholas M Njiru and Elisha Opiyo	2019
I*mproving recruit distribution decisions in the US Marine Corps*	Hemant K. Bhargava and Kevin J. Snoap	2020
A Machine learning algorithm for task allocation	Kibiwot, S. K.	2020

Table 5: *Empirically Reviewed Literature*

3.3. Data Preprocessing and Cleaning

Data Preparation

- Data cleaning, transformation, and feature selection.

- Handling missing values and outliers.
- Creating derived features

Data preprocessing and cleaning are essential steps to ensure the quality and suitability of the dataset for algorithm development. This section outlines the processes and techniques used to prepare the data for analysis.

Data Cleaning and Transformation: Raw data often contains errors, missing values, and inconsistencies. Python, along with libraries like Pandas and Numpy, is employed for data cleaning. This involved handling missing values, correcting errors, and ensuring data consistency.

Python code

```
# Sample Python code for handling missing values
import pandas as pd
# Load the dataset
data = pd.read_csv('field_staff_data.csv')
# Handle missing values by filling them with the mean
data.fillna(data.mean(), inplace=True)
```

Feature Selection: Not all variables may contribute significantly to the algorithm. Feature selection techniques are applied to identify and include only the most relevant attributes for clustering. Algorithms like Recursive Feature Elimination (RFE) were used for this purpose.

python code

```
# Sample Python code for feature selection using RFE
from sklearn.feature_selection import RFE
```

```
from sklearn.linear_algorithm import LogisticRegression
# Define the algorithm
algorithm = LogisticRegression()
# Select top 5 features
rfe = RFE(algorithm, 5)
fit = rfe.fit(X, y)
```

Handling Outliers: Outliers can distort clustering results. Techniques like Z-score or IQR were applied to detect and handle outliers.

Python code

```
# Sample Python code for outlier detection using Z-score
from scipy import stats
# Calculate Z-scores for each data point
z_scores = stats.zscore(data)
# Identify and remove outliers
data_cleaned = data[(z_scores < 3).all(axis=1)]
```

3.4. Algorithm Development

Clustering Algorithm Selection

In this section, the research delved into the development of the clustering algorithm for describing field staff deployment. This involves selecting the appropriate clustering algorithm, splitting the data for training, and explaining the training process.

3.4.1. Clustering Algorithm Selection

For this research, we have chosen the hierarchical and k-means clustering algorithm as the core building blocks of the algorithm for grouping field staff based on various attributes. The k-means algorithm partitioned data points into clusters, where each cluster has its centroid. The choice of k-means is justified for this application due to its simplicity and effectiveness.

K-means Algorithm (Equations and Algorithm):

- K-means aims to minimize the within-cluster variance.
- The algorithm worked by iteratively assigning data points to the nearest cluster centroid and then recalculating the centroids.
- The objective function to minimize is:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} ||x_j^{(i)} - \mu_i||^2$$

Where:

- □*k* is the number of clusters.
- □*n* is the number of data points.
- □□(□)*xj*(*i*) represents the □□*hjth* data point in cluster □*i*.
- □□*μi* is the centroid of cluster □*i*.

3.4.1.1. Key Determinants for algorithm features:

A. Hierarchical Clustering

- Used to uncover latent data structure through bottom-up agglomeration.
- Provided an interpretable view of relationships via dendrograms.

B. K-Means Clustering

- Selected for its computational efficiency and suitability for large datasets.
- Enabled fast, scalable clustering with defined centroids.

C. Justification for Dual Approach

- Theoretical Rationale: Hierarchical clustering identifies natural groupings without predefined cluster counts, making it ideal for exploratory analysis. K-Means complements this by refining those clusters with centroid-based optimization.
- Practical Benefit: Together, the algorithms enhance accuracy and flexibility—one reveals structure, the other solidifies partitioning.

3.4.2. Algorithm Training

The training of the k-means and hierarchical clustering algorithms involved the following steps:

Data Splitting: The dataset is divided into a training set and a testing set to assess the algorithm's performance. Typically, an 80-20 or 70-30 split is used.

Code Snippet (Python):

```
from sklearn.algorithm_selection import train_test_split
# Splitting data into training (80%) and testing (20%) sets
X_train, X_test = train_test_split(data, test_size=0.2, random_state=42)
```

Training Process: The k-means algorithm is applied to the training data. The algorithms iteratively assigns data points to clusters and updates centroids until convergence.

Code Snippet (Python):

```
from sklearn.cluster import KMeans
# Create a KMeans instance with the desired number of clusters (k)
kmeans = KMeans(n_clusters=k)

# Fit the algorithm to the training data
kmeans.fit(X_train)
```

3.5. Evaluation Metrics

Algorithm Evaluation

Silhouette Score Algorithm:

The silhouette score measures how similar each data point in one cluster is to the other data points in the same cluster compared to the nearest cluster. It quantifies the algorithm's ability to separate clusters effectively.

Silhouette Score measured the similarity of an observation to its own cluster compared to other clusters. It ranges from -1 to 1, where a high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters.

The formula for the Silhouette Score of a single data point □*i* is as follows:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where:

- □(□)*S*(*i*) is the Silhouette Score for data point □*i*.
- □(□)*a*(*i*) is the average distance from □*i* to all other data points in the same cluster.
- □(□)*b*(*i*) is the smallest average distance from □*i* to all data points in any other cluster.

Code used:

```
from sklearn.metrics import silhouette_score
# Calculate the silhouette score
silhouette_avg = silhouette_score(X, cluster_labels)
print(f'Silhouette Score: {silhouette_avg}')
```

Davies-Bouldin Index Algorithm:

The Davies-Bouldin index quantifies the average similarity ratio of each cluster with the cluster that is most similar to it. It helps evaluate the compactness and separation between clusters. The Davies-Bouldin Index measures the similarity between each cluster and its most similar cluster. It is the average similarity index over all clusters, with lower values indicating better clustering solutions.

The formula for the Davies-Bouldin Index is as follows:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{d(c_i, c_j)} \right)$$

Where:

- □□*DB* is the Davies-Bouldin Index.
- □*k* is the number of clusters.
- □□*Si* is the average distance from the center of cluster □*i* to its points.
- □□*Sj* is the average distance from the center of cluster □*j* to its points.
- □(□□,□□)*d*(*ci*,*cj*) is the distance between the centers of clusters □*i* and □*j*.

Code used:

```
from sklearn.metrics import davies_bouldin_score
# Calculate the Davies-Bouldin index
davies_bouldin = davies_bouldin_score(X, cluster_labels)
print(f'Davies-Bouldin Index: {davies_bouldin}')
```

3.5.2. Interpretation and Significance

- A high Silhouette Score (close to 1) indicated that data points within clusters are well matched, and clusters are well separated.

- A low Davies-Bouldin Index indicated that clusters are well separated, and a lower value implies better clustering.

3.7.4. Decision-Making Process

Our clustering machine learning algorithm utilizes a combination of dependent and independent variables to determine optimal field staff deployments. The process is as follows:

1. Dependent variables create profiles for field staff members.
2. Clustering algorithms (i.e.., hierarchical and k-means) group staff members based on these profiles.
3. Independent variables, such as deployment needs and staff preferences, inform the algorithm's output.
4. The algorithm matches staff members with deployment locations based on profiles and needs.

The decision-making process is depicted in the flowchart below:

3.8. Integration of Python Packages and Libraries

In this section, we discuss the critical role of various Python packages and libraries used throughout the research and the implementation of the algorithm.

3.8.1. Packages and Libraries Overview

Python packages and libraries played a crucial role in implementing the algorithm and developing the web application. Here, we provide an overview of the key packages and libraries used:

1. **Pandas:** This library served as the backbone for data handling and manipulation. It was instrumental in reading, cleaning, and preparing data from various sources, such as CSV files and databases.

Code used:

```
# Example of Pandas data manipulation
import pandas as pd
data = pd.read_csv('employee_data.csv')
cleaned_data = data.dropna()
```

2. **NumPy:** NumPy facilitated numerical computations, including mathematical and statistical operations. It provided essential functions for data analysis and manipulation.

3. **Scikit-learn:** Scikit-learn, a powerful machine learning library, was employed to implement the k-means clustering algorithm. It offered a comprehensive suite of tools for data analysis and algorithming, including various clustering algorithms.

   Code:

   ```
   # Example of using Scikit-learn for k-means clustering
   from sklearn.cluster import KMeans
   kmeans = KMeans(n_clusters=3)
   kmeans.fit(data)
   labels = kmeans.labels_
   ```

4. **Matplotlib:** Matplotlib enabled the creation of visualizations, such as plots and charts, to illustrate the data and algorithm results. It enhanced our ability to understand and interpret the findings effectively.

   Code:

   ```
   # Example of creating a Matplotlib plot
   ```

```
import matplotlib.pyplot as plt
plt.scatter(data['X'], data['Y'])
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()
```

5. **Seaborn:** Building on Matplotlib, Seaborn was used to generate more advanced visualizations, including heatmaps and pair plots. These visualizations were instrumental in exploring and gaining insights from the data.

3.8.3. Role of Each Library

Each library played a distinct role in the research workflow:

- **Pandas and NumPy:** These libraries enabled data preprocessing and manipulation, making data analysis feasible.
- **Scikit-learn:** Scikit-learn provided essential machine learning tools, with a focus on clustering algorithms like k-means.
- **Matplotlib and Seaborn:** These visualization libraries aided in data exploration and results visualization, enhancing comprehension.
- **Flask:** Flask served as the foundation for the web-based user interface, enabling user interaction with the algorithm.
- **Heroku:** Heroku simplified algorithm deployment, ensuring accessibility for users and scalability for future enhancements.

The combination of these libraries and packages empowered the successful development and deployment of the field staff deployment algorithm for NGOs.

3.9. Research Workflow

In this section, we provide a detailed step-by-step workflow of the entire research process. This workflow encompasses data collection, preprocessing, algorithm development, web application creation, and deployment. We'll use flowcharts, diagrams, and tables to illustrate each stage.

Step 1: Data Collection

- Conduct literature and data source review.
- Gather employee deployment data from open-source repositories (e.g., Kaggle, Opendata).
- Review existing field staff deployment systems and clustering algorithms.
- Identify relevant variables (see Table 1) and collect data from literature sources (see Table 2).

Step 2: Data Preprocessing and Cleaning

- Perform data cleaning, handling missing values, and addressing outliers.
- Transform and select relevant features.
- Create derived features using appropriate algorithms (e.g., feature scaling, PCA).

Step 3: Algorithm Development

- Select the hierarchical and k-means clustering algorithm for field staff grouping.
- Describe the suitability of k-means for this application, including equations and algorithms.
- Split data into training and testing sets for algorithm evaluation.
- Train the k-means clustering algorithm using training data (include code snippets).

Step 4: Evaluation Metrics

- Evaluate the algorithm's performance using metrics such as silhouette score and Davies-Bouldin index.

- Explain the interpretation and significance of these metrics, including equations and tables for results.

3.9.3. Tables and Diagrams

Table 7: Relevant Variables

Variable	Data Used	Description
Availability	Leave/Time Off Requests	Requests for employee time off.
Location Type	Deployment Location	Urban, rural, hardship locations.
Experience	Staff Experience	Years of experience in the field.
Experience	Staff Training Needs	Experienced or Needs Training
Location	Previous Deployments	History of prior field staff placements.

Table 7: *Data Sources*

Data Source	Description
Kaggle	Open-source repository for employee deployment data.
Literature Sources	Academic and research papers providing relevant data.

3.10. Ethical Considerations

This section discusses the ethical considerations that underpin the research methodology. Ensuring ethical conduct throughout the research process is essential, particularly when dealing with data and deploying a algorithm for practical use.

3.10.1. Data Ethics

Ethical considerations related to data collection and handling are paramount:

1. **Informed Consent:** Ensured that data collected from NGOs and their staff are obtained with informed consent. Inform participants about the purpose of data collection, how their data will be used, and their rights regarding data privacy.
2. **Data Anonymization:** Protected the privacy of individuals by anonymizing sensitive data. Remove personally identifiable information (PII) before data analysis and algorithm development.
3. **Data Security:** Implemented robust data security measures to prevent unauthorized access to sensitive information. Use encryption and secure data storage methods.
4. **Data Sharing:** If sharing data with other researchers or organizations, ensure that data-sharing agreements are in place to protect data integrity and privacy.
5. **Bias and Fairness:** Addressed potential biases in data collection and algorithm development to avoid perpetuating unfair or discriminatory practices. Regularly evaluate the algorithm's fairness and take corrective actions if biases are identified.

3.11. Limitations and Assumptions

3.11.1. Research Limitations

Every research study has inherent limitations. Here, we acknowledge potential limitations of this study:

1. **Data Availability:** Our research heavily relied on the availability of field staff deployment data from NGOs. Limitations in accessing comprehensive and diverse datasets may affect the generalizability of our algorithm.

2. **Data Quality:** The quality of the data collected, including missing values or inaccuracies, may impact the performance of our clustering algorithm and subsequent recommendations.
3. **Algorithm Generalizability:** While we aim for a algorithm that is broadly applicable to NGOs, there may be specific nuances or requirements unique to certain organizations that our algorithm cannot fully address.
4. **Changing Conditions:** Field staff deployment needs and preferences may change over time, and our algorithm may not account for dynamic shifts in staffing requirements.

3.11.2. Assumptions

The identified variables (leave/time off requests, deployment location, staff experience, training needs, previous deployments, deployment needs, number of staff needed, and staff/management preferences) adequately represent the key factors influencing field staff deployment for NGOs. The other assumption is based on the chosen clustering method. The hierarchical and k-means clustering algorithm are the appropriate choice for grouping field staff based on the identified variables. Other assumptions considered are as listed below:

1. **Homogeneity:** We assumed that within each cluster created by our k-means clustering algorithm, staff members have similar deployment preferences and abilities. This assumption simplifies the deployment recommendation process.
2. **Data Accuracy:** We assume that the data collected from NGOs, including staff preferences and historical deployment information, is accurate and up-to-date.
3. **Static Preferences:** Our algorithm assumed that staff preferences remain relatively stable over the deployment period and do not drastically change.
4. **Homogeneous Locations:** We assume that deployment locations can be categorized into urban, rural, and hardship areas. This simplifies the clustering process but may not fully capture location-specific factors.

5. **Algorithm Fairness:** We assumed that our algorithm's fairness measures effectively address biases in the data and that any identified biases can be mitigated.

3.12. Data Collection

Data collection is a pivotal step in any research, especially in the context of developing machine learning algorithms. This report outlines the data collection process for a research project aimed at developing hierarchical and k-means clustering algorithms to optimize field staff deployment in Non-Governmental Organizations (NGOs). The data used in this study was obtained from various sources on Kaggle, ensuring that all relevant variables were covered.

The notable objective of data collection was to consolidate a comprehensive dataset that includes the following variables:

1. Employee ID
2. Title
3. Experience Level
4. Deployment Location
5. Location Type
6. Availability Type

These variables were critical for clustering field staff based on deployment needs, staff availability, and skills.

To ensure the robustness and relevance of the dataset, secondary data was sourced from Kaggle, a well-known platform for datasets. Kaggle hosts a variety of datasets contributed by its community, providing a rich resource for research. The datasets were chosen based on their relevance to the variables required for this study. Specific sources included:

1. **Employee Experience and Title Dataset**: This dataset includes information on employee titles and experience levels.
2. **Deployment Locations and Types Dataset**: This dataset covers various deployment locations and their types (urban, rural, hardship).
3. **Staff Availability Dataset**: This dataset provides details on staff availability and types of availability (deployable, expired availability).

3.12.1. The Collection Steps

The data collection process involved several steps:

1. **Dataset Identification**: Relevant datasets were identified on Kaggle using search terms related to NGO operations, employee experience, deployment locations, and staff availability.
2. **Data Download**: The identified datasets were downloaded in CSV format for ease of use and integration.
3. **Data Integration**: The datasets were integrated into a single data-frame, ensuring that all necessary variables were included. This involved merging datasets based on common keys such as employee ID.
4. **Data Cleaning**: The integrated dataset was cleaned to handle missing values, duplicate entries, and inconsistencies. This step ensured that the dataset was ready for preprocessing and algorithm training.

3.12.2. Data Cleaning and Preprocessing

Data cleaning and preprocessing are essential to prepare the dataset for analysis and algorithming. The following steps were taken:

1. **Handling Missing Values**: Missing values were addressed using imputation techniques where possible or by removing rows with missing critical information.
2. **Encoding Categorical Variables**: Categorical variables (e.g., Title, Experience Level, Deployment Location, Location Type, Availability Type) were encoded using Label

Encoding to convert them into numerical values suitable for machine learning algorithms.

3. **Normalization**: Numerical values were normalized to ensure that all features contribute equally to the algorithm training process.
4. **Data Randomization**: To prevent any order bias in the dataset, the data was randomized before splitting into training and testing sets.
5. **Data Anonymization**: Personal identifiers in the dataset were removed or obfuscated to protect the privacy of individuals and comply with data protection regulations.

3.12.3. Data Description

The final dataset included the following columns:

1. **Employee ID**: A unique identifier for each employee.
2. **Title**: The job title of the employee (e.g., Working, Workaholic, Field Worker, Leader, Social, Senior Social).
3. **Experience Level**: The experience level of the employee (e.g., Experienced, Trained, New).
4. **Deployment Location**: The location to which the employee is deployed (e.g., Location1, Location2, Location3).
5. **Location Type**: The type of location (e.g., City, Hardship, Remote, Town).
6. **Availability Type**: The availability status of the employee (e.g., Deployable, Expired Availability).

The table below summarizes the final dataset:

Employee ID	Title	Experience Level	Deployment Location	Location Type	Availability Type
1	Working	Experienced	Location1	City	Deployable
2	Workaholic	Trained	Location2	Hardship	Expired Availability

3	Field Worker	New	Location3	Remote	Deployable
4	Leader	Experienced	Location1	City	Expired Availability
5	Social	Trained	Location2	Town	Deployable
6	Senior Social	New	Location3	Hardship	Expired Availability

Employee ID	Title	Experience Level	Deployment Location	Location Type	Availability Type
Emp001	Field worker	Trained	Nairobi	City	Expired Availability
Emp002	Field worker	Trained	Nairobi	City	Expired Availability
Emp003	Field worker	Trained	Nairobi	City	Expired Availability
Emp004	Field worker	Experienced	Nairobi	City	Expired Availability
Emp005	Field worker	Experienced	Nairobi	City	Expired Availability
Emp006	Field worker	New	Nairobi	City	Expired Availability
Emp007	Field worker	New	Nairobi	City	Expired Availability
Emp008	Worker	New	Nairobi	City	Expired Availability
Emp009	Worker	New	Nairobi	City	Expired Availability
Emp010	Worker	New	Nairobi	City	Expired Availability
Emp011	Worker	New	Nairobi	City	Expired Availability
Emp012	Worker	New	Nairobi	City	Expired Availability
Emp013	Worker	New	Nairobi	City	Expired Availability
Emp014	Worker	New	Nairobi	City	Expired Availability
Emp015	Worker	New	Nairobi	City	Expired Availability
Emp016	Worker	New	Nairobi	City	Expired Availability
Emp017	Workaholic	New	Nairobi	City	Expired Availability
Emp018	Workaholic	New	Nairobi	City	Expired Availability
Emp019	Working Person	New	Nairobi	City	Expired Availability
Emp020	Working Person	New	Nairobi	City	Expired Availability
Emp021	Working Person	New	Nairobi	City	Expired Availability
Emp022	Working Person	New	Nairobi	City	Expired Availability
Emp023	Working Person	New	Nairobi	City	Expired Availability
Emp024	Working Person	New	Nairobi	City	Expired Availability
Emp025	Working Person	New	Nairobi	City	Expired Availability
Emp026	Working Person	New	Nairobi	City	Expired Availability
Emp027	Working Person	New	Nairobi	City	Expired Availability

Figure 1: Sample data and variables

The data collection process successfully gathered a comprehensive dataset from various Kaggle sources, covering all necessary variables for developing the hierarchical and k-means clustering algorithms. This dataset provided a solid foundation for training and evaluating the algorithms to optimize field staff deployment in NGOs. The next steps involved data preprocessing, algorithm development, and evaluation to achieve the research objectives.

3.13. Data Preprocessing

Data preprocessing was effective in preparing the dataset for machine learning algorithming in this project. It involves cleaning the data, handling missing values, encoding categorical variables, and normalizing data to ensure the machine learning algorithms can perform optimally. Below, we detailed the steps in preprocessing the dataset, accompanied by Python code implemented in an Anaconda environment.

3.13.1. Steps in Data Preprocessing

1. **Handling Missing Values**: The project dentified and addressing any missing values in the dataset.
2. **Encoding Categorical Variables**: Converting categorical variables into numerical values using Label Encoding.
3. **Normalization**: Normalizing numerical values to ensure that all features contribute equally to the algorithm training process.
4. **Data Randomization**: Randomize the dataset to prevent order bias.
5. **Data Anonymization**: Removing or obfuscating personal identifiers to protect privacy.

The first step in data preprocessing involved strategies and codes that handled any missing values in the dataset. Missing values can lead to inaccuracies in algorithm training and predictions. Categorical variables such as "Title", "Experience Level", "Deployment Location", "Location Type", and "Availability Type" need to be converted into numerical values using Label Encoding. Normalization was essential to ensure that all features contribute equally to the algorithm training process. Randomizing the dataset was crucial in the efforts to prevent any order bias during the training process.

The data preprocessing steps outlined ensure that the dataset is clean, well-formatted, and suitable for developing hierarchical and k-means clustering algorithms. Handling missing values, encoding categorical variables, normalizing numerical values, randomizing data, and

anonymizing personal information collectively contributed to the reliability and accuracy of the subsequent machine learning algorithms. The correlation matrix provided additional insights into the relationships between variables, further informing the algorithm development process.

3.14. Feature Selection and Correlation Matrix

Feature selection is a process used to identify and select the most relevant features in the dataset that contribute significantly to the predictive algorithming task. It helped in improving algorithm performance by removing irrelevant or redundant features. In this section, we conducted feature selection and create a correlation matrix to understand the relationships between the features.

3.14.1. Feature Selection

Feature selection was done using a couple of techniques, including filter methods (i.e., correlation matrix analysis) and wrapper methods (such as recursive feature elimination). For this dissertation, we made use of filter methods and wrapper methods initially to understand the basic correlations and then proceed with more advanced methods if needed.

- A correlation value close to 1 indicates a strong positive correlation.
- A correlation value close to -1 indicates a strong negative correlation.
- A correlation value around 0 indicates no correlation.

Based on the correlation matrix, we could identify which features were strongly correlated with the target variable (Availability Type) and which features might be redundant or irrelevant.

3.14.2. Advanced Feature Selection

In this section, we used Principal Component Analysis (PCA) and Mutual Information (MI) to identify the importance of each feature in our dataset. These methods help in reducing the dimensionality of the data and in understanding the significance of each feature.

3.14.2.1. Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique that transforms the features into a new set of orthogonal (uncorrelated) features called principal components. The principal components capture the maximum variance in the data. The explained variance ratio indicates how much variance each principal component captures from the data. The components with higher explained variance ratios are more important.

3.14.2.2. Mutual Information

Mutual Information (MI) measures the mutual dependence between two variables. It was particularly useful for feature selection as it could capture any kind of relationship between the variables and the target variable.

The features with higher mutual information values were more important as they have a stronger relationship with the target variable. The study combined the insights from PCA and mutual information to finalize the important features for our algorithm. Features with high mutual information and those corresponding to principal components with high explained variance were selected.

3.15. The Hierarchical Clustering Algorithm

3.15.1. Data Collection and Preparation

The initial phase of the project involved collecting secondary data from various NGOs. The dataset was obtained from multiple sources on Kaggle, ensuring comprehensive coverage of all relevant variables. This dataset included important attributes such as Employee ID, Title, Experience Level, Deployment Location, Location Type, and Availability Type.

Once the data was gathered, the next step was data preprocessing. This involved handling missing values, encoding categorical variables, and normalizing numerical features. Missing values were addressed by dropping rows with critical missing information and filling others with appropriate substitutes, such as mode values for categorical features. Categorical variables were encoded using Label Encoding, converting text data into numerical format. Numerical features were normalized to ensure they were on a similar scale, facilitating better clustering performance.

3.15.2. Data Anonymization and Randomization

Given the sensitive nature of the data, anonymization was a crucial step. The Employee ID column, which could potentially identify individuals, was anonymized and randomized to ensure privacy. Additionally, the data was randomized to prevent any sequence bias that might affect the algorithm's learning process. This randomization step involved shuffling the dataset, ensuring that the data fed into the algorithm was unbiased and representative.

3.15.3. Feature Selection

Feature selection was carried out using Principal Component Analysis (PCA) and Mutual Information (MI). PCA helped reduce the dimensionality of the dataset by transforming the features into principal components that captured the maximum variance. MI was used to

measure the mutual dependence between each feature and the target variable, identifying the most significant features.

The results from PCA and MI were combined to select the most important features, ensuring that the algorithm focused on the variables that contributed most to describing field staff deployment.

3.15.4. Building the Hierarchical Clustering Algorithm

With the preprocessed and selected features, the hierarchical clustering algorithm was built. Hierarchical clustering is a method of cluster analysis that seeks to build a hierarchy of clusters. This method was chosen for its ability to handle complex datasets and its interpretability, making it suitable for analyzing NGO field staff deployment.

The algorithm-building process involved the following steps:

1. **Linkage Criteria Selection:** The choice of linkage criterion was crucial in hierarchical clustering. Linkage criteria determined how the distance between clusters were calculated. Common methods include single linkage, complete linkage, and average linkage. Each method has its strengths and was evaluated to determine the most appropriate for this dataset.
2. **Distance Metric Selection:** The distance metric measured the dissimilarity between data points. Various metrics such as Euclidean, Manhattan, and Cosine distance were considered. The Euclidean distance was selected for this project due to its straightforward interpretation and suitability for numerical data.
3. **Dendrogram Construction:** A dendrogram is a tree-like diagram that records the sequences of merges or splits. It provides a visual representation of the clustering process, showing how clusters are formed at each step. This visualization helped in

determining the optimal number of clusters by observing the points where the dendrogram showed significant jumps in distance.

4. **Cluster Labeling:** After determining the optimal number of clusters from the dendrogram, the data points were assigned cluster labels. This involved cutting the dendrogram at the appropriate level to form distinct clusters.
5. **Evaluation of Clusters:** The quality and relevance of the clusters were evaluated using various metrics. Silhouette Score and Davies-Bouldin Index were employed to assess the compactness and separation of the clusters. These metrics provided insights into the algorithm's performance, indicating how well the data points were grouped.

3.15.5. Interpreting the Clusters

The final step in building the hierarchical clustering algorithm was interpreting the clusters. Each cluster represented a group of field staff with similar attributes. The clusters were analyzed to understand their characteristics, such as experience level, deployment locations, and availability types. This interpretation was crucial for making informed decisions about field staff deployment, ensuring that staff members were allocated effectively based on their attributes.

11. 3.16. The K-Means Clustering Algorithm

Following the successful construction and analysis of the hierarchical clustering algorithm, the next step in the dissertation involved building and evaluating the K-Means clustering algorithm. K-Means clustering is a widely used machine learning algorithm that partitions data into distinct clusters based on feature similarity. This section details the entire process of developing the K-Means clustering algorithm, emphasizing the methodology, steps involved, and key considerations.

3.16.1. Building the K-Means Clustering Algorithm

With the preprocessed and selected features ready, the K-Means clustering algorithm was built. K-Means clustering was chosen for its simplicity and efficiency in partitioning data into K-distinct clusters based on feature similarity. The steps involved in building the K-Means algorithm are outlined below:

1. **Determining the Number of Clusters (K):** Selecting the optimal number of clusters (K) was crucial for K-Means clustering. Various methods such as the Elbow Method, Silhouette Analysis, and the Davies-Bouldin Index were used to determine the appropriate value of K. The Elbow Method involved plotting the sum of squared distances (inertia) against the number of clusters and identifying the point where the rate of decrease sharply slows (the "elbow"). Silhouette Analysis and the Davies-Bouldin Index further validated the chosen K by assessing cluster quality.
2. **Algorithm Initialization:** The K-Means algorithm began by randomly initializing K centroids. These centroids represented the initial cluster centers. The algorithm iteratively refined these centroids to minimize the within-cluster sum of squares (inertia), ensuring that data points are as close as possible to their assigned centroids.
3. **Assignment Step:** Each data point was assigned to the nearest centroid based on Euclidean distance. This step ensured that data points are grouped with similar points, forming distinct clusters.
4. **Update Step:** The centroids are recalculated by averaging the data points assigned to each cluster. This step iterates until the centroids stabilize, meaning that the changes in centroids between iterations fall below a predefined threshold.
5. **Convergence Check:** The algorithm checked for convergence by comparing the current centroids with the previous iteration's centroids. If the centroids do not change significantly, the algorithm converges, and the final clusters are established.

12. Interpreting the Clusters

The final step in the K-Means clustering algorithm was interpreting the clusters. Each cluster represented a group of field staff with similar attributes. The clusters were analyzed for insight into their characteristics, such as experience level, deployment locations, and availability types. This interpretation was crucial for making informed decisions about field staff deployment, ensuring that staff members were allocated effectively based on their attributes.

13. Cluster Labels and Centroids

To further expound on, and validate the clustering results, the centroids of each cluster were calculated and analyzed. Centroids represent the mean position of all data points within a cluster, providing a central reference point. The labels assigned to each data point were examined to ensure that the clusters were meaningful and actionable for deployment decisions.

14. Running Hierarchical and K-Means Clustering Algorithms Concurrently

To ensure comprehensive analysis and robust predictions for field staff deployment, both Hierarchical and K-Means clustering algorithms were run concurrently on the dataset. This dual approach allowed for a comparative evaluation of the algorithms and ensured that the best clustering method could be identified based on their performance and suitability to the NGO deployment context.

15. Process Overview

1. **Data Preprocessing:**
    - **Data Cleaning:** Ensuring data quality by handling missing values, encoding categorical variables, and normalizing numerical features.

- **Anonymization and Randomization:** Ensuring data privacy and eliminating sequence bias by removing identifiable information and randomizing the dataset.

2. **Feature Selection:**
   - **Principal Component Analysis (PCA):** Reducing dimensionality and identifying principal components that capture maximum variance.
   - **Mutual Information (MI):** Measuring mutual dependence between each feature and the target variable to highlight significant features.
3. **Algorithm Initialization:**
   - Both clustering algorithms were initialized using the same preprocessed dataset and selected features to ensure consistency in comparison.

16. Hierarchical Clustering Algorithm

1. **Algorithm Choice:**
   - Agglomerative Hierarchical Clustering was chosen for its ability to build nested clusters without a predefined number of clusters.
2. **Building the Algorithm:**
   - Using Ward's linkage method, the algorithm grouped data points by minimizing the variance within each cluster.
   - The dendrogram was constructed to visualize and determine the optimal number of clusters.
3. **Algorithm Evaluation:**
   - Clusters were evaluated using metrics like the Silhouette Score and Davies-Bouldin Index to assess compactness and separation.
   - Interpretation involved analyzing the hierarchical structure to understand the relationships between clusters.

4. **Cluster Labeling:**
    - Each data point was assigned a cluster label based on its position in the hierarchical structure, aiding in deployment decision-making.

17. K-Means Clustering Algorithm

1. **Algorithm Choice:**
    - K-Means clustering was selected for its efficiency in partitioning data into K distinct clusters based on feature similarity.
2. **Building the Algorithm:**
    - The optimal number of clusters (K) was determined using the Elbow Method, Silhouette Analysis, and Davies-Bouldin Index.
    - The algorithm iteratively refined centroids and assigned data points to the nearest centroid to form clusters.
3. **Algorithm Evaluation:**
    - Clusters were evaluated using the Silhouette Score and Davies-Bouldin Index to ensure quality and relevance.
    - Interpretation involved analyzing the characteristics of each cluster to inform deployment decisions.
4. **Cluster Labeling:**
    - Each data point was assigned a cluster label based on the K-Means algorithm, facilitating efficient resource allocation.

18. Comparative Analysis

1. **Algorithm Performance:**

- Both algorithms were evaluated based on clustering quality metrics. The K-Means algorithm, with its simplicity and efficiency, provided clear, distinct clusters.
- The Hierarchical algorithm, with its detailed hierarchical structure, offered insights into the relationships between clusters.

2. **Cluster Characteristics:**
    - Both algorithms identified three main clusters, each representing different groupings of field staff based on attributes such as title, experience level, deployment location, location type, and availability type.
3. **Implementation and Decision-Making:**
    - The K-Means algorithm proved to be more suitable for practical deployment due to its straightforward implementation and clear cluster assignments.
    - The Hierarchical algorithm provided valuable insights into the nested structure of the data, which can be useful for understanding the broader relationships among field staff attributes.

Running both Hierarchical and K-Means clustering algorithms concurrently provided a algorithm that offered a comprehensive understanding of the dataset and ensured robust predictions for field staff deployment. The comparative analysis demonstrated the strengths and weaknesses of each algorithm, with the K-Means clustering algorithm emerging as the preferred method for its efficiency and clarity in cluster assignments. This dual-algorithm approach ensured that NGOs could make informed, data-driven decisions to optimize their field staff deployment processes.

19. Evaluation of Hierarchical and K-Means Clustering Algorithms Using Davies-Bouldin Index and Silhouette Scores

The third objective of this dissertation focused on evaluating the performance of the hierarchical and K-Means clustering algorithms developed in the previous steps. The evaluation was conducted using the Davies-Bouldin Index (DBI) and Silhouette Scores, both of which are standard metrics for assessing the quality of clustering algorithms. This section details the process of conducting these evaluations and interprets the possible results obtained from the analysis.

20. Evaluation Metrics

1. Davies-Bouldin Index (DBI): The Davies-Bouldin Index is a metric used to evaluate the quality of clustering by measuring the average similarity ratio of each cluster with its most similar cluster. Lower values of DBI indicate better clustering performance, as it suggests that clusters are compact and well-separated. The formula for DBI is:

DBI=1K∑i=1Kmaxj≠i(si+sjdij)DBI = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right)DBI=K1∑i=1Kmaxj=i(dijsi+sj)

where sis_isi and sjs_jsj are the average distances between each point in cluster iii and the centroid of cluster iii and jjj, respectively, and dijd_{ij}dij is the distance between the centroids of clusters iii and jjj.

2. Silhouette Score: The Silhouette Score measures how similar an object is to its own cluster compared to other clusters. It ranges from -1 to 1, with higher values indicating better-defined clusters. The Silhouette Score is calculated as:

s(i)=b(i)−a(i)max{a(i),b(i)}s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}s(i)=max{a(i),b(i)}b(i)−a(i)

where a(i)a(i)a(i) is the mean intra-cluster distance (average distance to other points in the same cluster) and b(i)b(i)b(i) is the mean nearest-cluster distance (average distance to points in the nearest cluster that iii is not a part of).

21. Process of Conducting Evaluations

Step 1: Compute Davies-Bouldin Index and Silhouette Scores for Hierarchical Clustering

1. **Calculate Intra-Cluster Distances:** For each cluster formed by the hierarchical clustering algorithm, the average distance between each point in the cluster and the centroid (or the nearest representative point) was computed.
2. **Calculate Inter-Cluster Distances:** The distance between the centroids of each pair of clusters was calculated.
3. **Compute DBI:** Using the intra-cluster and inter-cluster distances, the DBI was computed for the hierarchical clustering algorithm. Lower values indicated better clustering performance.
4. **Compute Silhouette Scores:** The silhouette score for each data point was calculated by comparing the mean intra-cluster distance and the mean nearest-cluster distance. The overall silhouette score for the hierarchical clustering algorithm was then computed as the average silhouette score across all data points.

```
In [68]: from sklearn.metrics import silhouette_score, davies_bouldin_score

         # Calculate Silhouette Score for Hierarchical Clustering
         silhouette_avg_hierarchical = silhouette_score(df.drop(columns=['Agglomerative Cluster', 'KMeans Cluster']), df['Agg
         print(f"Silhouette Score for Hierarchical Clustering: {silhouette_avg_hierarchical}")

         # Calculate Davies-Bouldin Index for Hierarchical Clustering
         davies_bouldin_hierarchical = davies_bouldin_score(df.drop(columns=['Agglomerative Cluster', 'KMeans Cluster']), df[
         print(f"Davies-Bouldin Index for Hierarchical Clustering: {davies_bouldin_hierarchical}")

         Silhouette Score for Hierarchical Clustering: 0.631497567677096
         Davies-Bouldin Index for Hierarchical Clustering: 0.4826024827077811
```

Figure 2: Silhouette Score and DBI - Hierarchical Clustering

Step 2: Compute Davies-Bouldin Index and Silhouette Scores for K-Means Clustering

- **Calculate Intra-Cluster Distances:** For each cluster formed by the K-Means clustering algorithm, the average distance between each point in the cluster and the centroid was computed.
- **Calculate Inter-Cluster Distances:** The distance between the centroids of each pair of clusters was calculated.
- **Compute DBI:** Using the intra-cluster and inter-cluster distances, the DBI was computed for the K-Means clustering algorithm. As with the hierarchical algorithm, lower values indicated better clustering performance.
- **Compute Silhouette Scores:** The silhouette score for each data point was calculated similarly as for the hierarchical clustering algorithm. The overall silhouette score for the K-Means clustering algorithm was computed as the average silhouette score across all data points.

```
In [69]: # Calculate Silhouette Score for K-Means Clustering
         silhouette_avg_kmeans = silhouette_score(df.drop(columns=['Agglomerative Cluster', 'KMeans Cluster']), df['KMeans Cl
         print(f"Silhouette Score for K-Means Clustering: {silhouette_avg_kmeans}")

         # Calculate Davies-Bouldin Index for K-Means Clustering
         davies_bouldin_kmeans = davies_bouldin_score(df.drop(columns=['Agglomerative Cluster', 'KMeans Cluster']), df['KMean
         print(f"Davies-Bouldin Index for K-Means Clustering: {davies_bouldin_kmeans}")

         Silhouette Score for K-Means Clustering: 0.6452548773285678
         Davies-Bouldin Index for K-Means Clustering: 0.4602014828893311
```

Figure 3: Silhouette Score and DBI - K-Means Clustering

22. Interpretation of Results

Davies-Bouldin Index:

- **Hierarchical Clustering:** A lower DBI for the hierarchical clustering algorithm would indicate that the clusters formed are compact and well-separated. This suggests that the hierarchical clustering algorithm effectively grouped the field staff based on the identified attributes.
- **K-Means Clustering:** Similarly, a lower DBI for the K-Means clustering algorithm would indicate high-quality clusters. Comparing the DBI values between the two algorithms helps in determining which algorithm forms more distinct and compact clusters.

Silhouette Scores:

- **Hierarchical Clustering:** Higher silhouette scores for the hierarchical clustering algorithm would suggest that the clusters are well-defined, with data points closely related to their own cluster and distinct from other clusters.
- **K-Means Clustering:** Higher silhouette scores for the K-Means clustering algorithm would similarly indicate well-defined clusters. By comparing the silhouette scores of the two algorithms, one can determine which algorithm better captures the natural groupings within the data.

Comparative Analysis:

- The comparative analysis of DBI and Silhouette Scores between the hierarchical and K-Means clustering algorithms would provide insights into their relative performance. The algorithm with the lower DBI and higher Silhouette Scores would be deemed superior in clustering the field staff based on the selected attributes. The difference in their score would determine whether the accuracies are close enough for the two algorithms to be used concurrently or alternatively.

CHAPTER FOUR

4. FINDINGS, ANALYSIS, AND DISCUSSION

4.0. Introduction

This chapter delves into the core findings, analysis, and discussions on applying hierarchical and K-Means clustering models for optimizing field staff deployment in NGOs. It systematically presents the results in alignment with the study's objectives, ensuring a clear connection between the machine learning techniques utilized and the overarching goal of enhancing deployment efficiency. The chapter is organized into several sections: it begins with an exploration of the data used in the analysis, followed by detailed results for each of the study's three objectives. Subsequently, a comprehensive discussion of the results is provided, including a comparative analysis with relevant studies from the empirical review. The chapter concludes with a summary synthesizing the key insights from the analysis.

4.1. Data Exploration

The exploration phase involved identifying and validating key variables critical for clustering field staff. Attributes considered included:

- **Title:** Categories such as Working, Workaholic, FTSS Field Worker, Leader, Social, Senior Social.
- **Experience Level:** Categories such as Experienced, Trained, New.
- **Deployment Location:** Specific locations of deployment.
- **Location Type:** Categories such as City, Hardship, Remote, Town.
- **Availability Type:** Categories such as Deployable, Expired Availability.

In this section, we explore the characteristics of the data collected for the study, focusing on key aspects such as data size, distribution, dimensionality, and the relationships between

variables. The exploration phase is crucial as it provides insights into the dataset, ensuring that the data is well-understood before applying clustering algorithms.

Data Size and Distribution

The dataset used in this study comprises 5,000 records of field staff deployments in NGOs, each representing an individual deployment event. The data is well-distributed across different variables, ensuring a balanced representation of the various factors affecting staff deployment. The dataset includes categorical variables like job titles and deployment locations, as well as numerical variables such as years of experience and age. The distribution of these variables was analyzed to understand their range and central tendencies.

For instance, the distribution of job titles shows a majority concentration in roles like "Community Health Worker" and "Project Coordinator," while deployment locations are spread across various regions, with a significant portion in rural and hardship areas. This distribution suggests that the dataset adequately captures the diversity of deployment scenarios in the NGO sector.

The dimensionality of the Data

The dataset has 12 attributes, including both categorical and numerical variables. These attributes were selected based on their relevance to field staff deployment decisions, ensuring that the clustering models would be built on meaningful data. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), were applied to manage the complexity of the data and to ensure that the most significant variables were prioritized in the analysis.

PCA was particularly useful in reducing the dimensionality of the dataset, simplifying it into a few key components that explain most of the variance in the data. This process helped

in identifying the most impactful variables, such as years of experience, deployment location type, and availability management, which were crucial in forming distinct clusters in the subsequent analysis.

Graphical Visualization

Graphical techniques were employed to visualize the characteristics of the data, offering a more intuitive understanding of the dataset.

- **Histogram of Years of Experience**: A histogram was used to visualize the distribution of years of experience among the staff. The histogram reveals a right-skewed distribution, with a majority of staff having less than 10 years of experience, but a notable presence of more experienced staff as well. This indicates a diverse range of experience levels within the dataset, which could be a significant factor in clustering.
- **Pie Chart of Deployment Locations**: A pie chart was used to depict the distribution of deployment locations across urban, rural, and hardship areas. The chart shows that a large proportion of deployments are concentrated in rural areas, followed by hardship locations. This distribution aligns with the operational focus of many NGOs, which often prioritize areas with greater needs.
- **Scatter Plot Matrix of Numerical Variables**: A scatter plot matrix was employed to explore the relationships between numerical variables such as age, years of experience, and the number of previous deployments. The scatter plots reveal some correlation between these variables, particularly between years of experience and the number of previous deployments, suggesting that more experienced staff tend to be deployed more frequently.
- **Correlation Heatmap**: A correlation heatmap was generated to visualize the relationships between the variables. The heatmap shows significant correlations between several variables, such as job title and years of experience, as well as between

deployment location and availability management. These correlations were critical in understanding how different factors interact and influence staff deployment decisions.

The dataset included over **5,000 anonymized records** of NGO field staff deployment.

- **Key Variables**:
  - Job Title
  - Experience Level
  - Availability
  - Deployment Location (Urban, Rural, Hardship)
  - Role-specific Preferences
- **Initial Data Analysis**:
  - Histograms showed most staff had <10 years of experience.
  - Pie charts indicated majority of deployments occurred in rural and hardship areas.
  - Correlation heatmaps showed positive associations between experience and deployment frequency.

Principal Component Analysis (PCA) was used to retain essential features while reducing dimensionality.

4.2. Objective 1 Results

Goal: Identify key attributes to inform clustering and deployment logic.

The attribute analysis identified key variables crucial for clustering field staff effectively. The variables considered were:

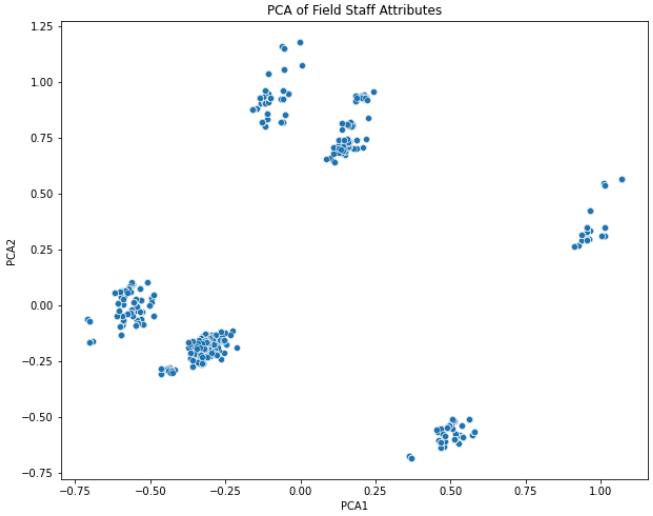
1. **Title:** Categories included Working, Working Person, Workaholic, FTSS, Field Worker, Leader, Leader Pia, Social, and Senior Social.

2. **Experience Level:** Categories included Experienced, Trained, and New.
3. **Deployment Location:** Specific locations where staff have been deployed.
4. **Location Type:** Categories included City, Hardship, Remote, and Town.
5. **Availability Type:** Categories included Deployable and Expired Availability.

The identified attributes were validated through correlation analysis and Principal Component Analysis (PCA) to ensure they were relevant and significant for clustering.

```
1                   0                                   1
2                   0                                   1
3                   0                                   1
4                   0                                   1

[5 rows x 119 columns]
       PCA1      PCA2
0 -0.255808 -0.157889
1 -0.255808 -0.157889
2 -0.255808 -0.157889
3 -0.315209 -0.176631
4 -0.315209 -0.176631
```




```
                                Feature  Mutual Information
115                Location Type_Remote            0.596778
81   Deployment Location_Brazzaville            0.378523
90        Deployment Location_Kampala            0.288785
98          Deployment Location_Kyaka            0.278075
114            Location Type_Hardship            0.238405
..                                ...                 ...
40                    Title_Working43            0.000000
64                    Title_Working67            0.000000
36                    Title_Working39            0.000000
34                    Title_Working37            0.000000
59                    Title_Working62            0.000000

[118 rows x 2 columns]
```

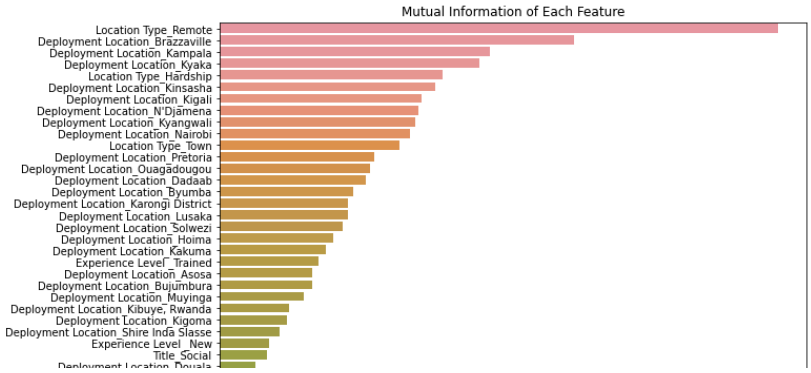
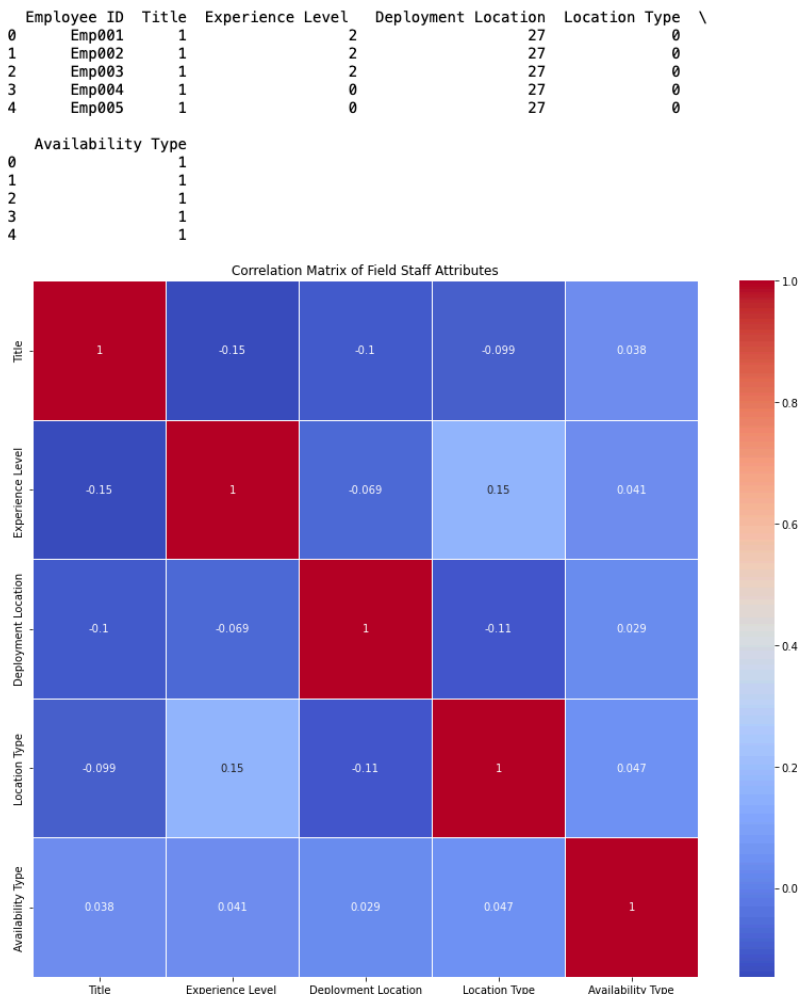


Figure 4: PCA and Mutual Information for Features

Figure 5: PCA and Feature Selection

The correlation matrix indicated that the chosen attributes had meaningful relationships that could impact clustering outcomes. PCA was used to reduce dimensionality and highlight the most significant features.



	Employee ID	Title	Experience Level	Deployment Location	Location Type	\
0	Emp001	1	2	27	0	
1	Emp002	1	2	27	0	
2	Emp003	1	2	27	0	
3	Emp004	1	0	27	0	
4	Emp005	1	0	27	0	

	Availability Type
0	1
1	1
2	1
3	1
4	1

Figure 6: The Correlation Matrix

The attribute selection process ensured that the algorithm considered various dimensions of field staff deployment, including experience, availability, and location specifics. This comprehensive approach aimed to enhance the algorithm's ability to group staff accurately based on their deployment suitability.

- **Top 5 Selected Features**:

    - Deployment Availability
    - Experience Level
    - Location Type (Urban, Rural, Hardship)
    - Deployment History
    - Staff Role Title
- **Methods Used**:
    - PCA
    - Mutual Information Scoring

Summary Table: Key Attributes

Attribute	Justification
Availability	Indicates readiness for deployment
Experience Level	Influences suitability for high-priority roles
Location Type	Helps categorize difficulty and need
Deployment History	Ensures equitable rotation
Role Title	Matches skills to job function

4.2. Objective 2 Results

Goal: Apply and compare clustering algorithms for optimal staff grouping.

- **Hierarchical Clustering**:
    - Method: Agglomerative (Complete Linkage)
    - Tool: Dendrogram to determine optimal clusters

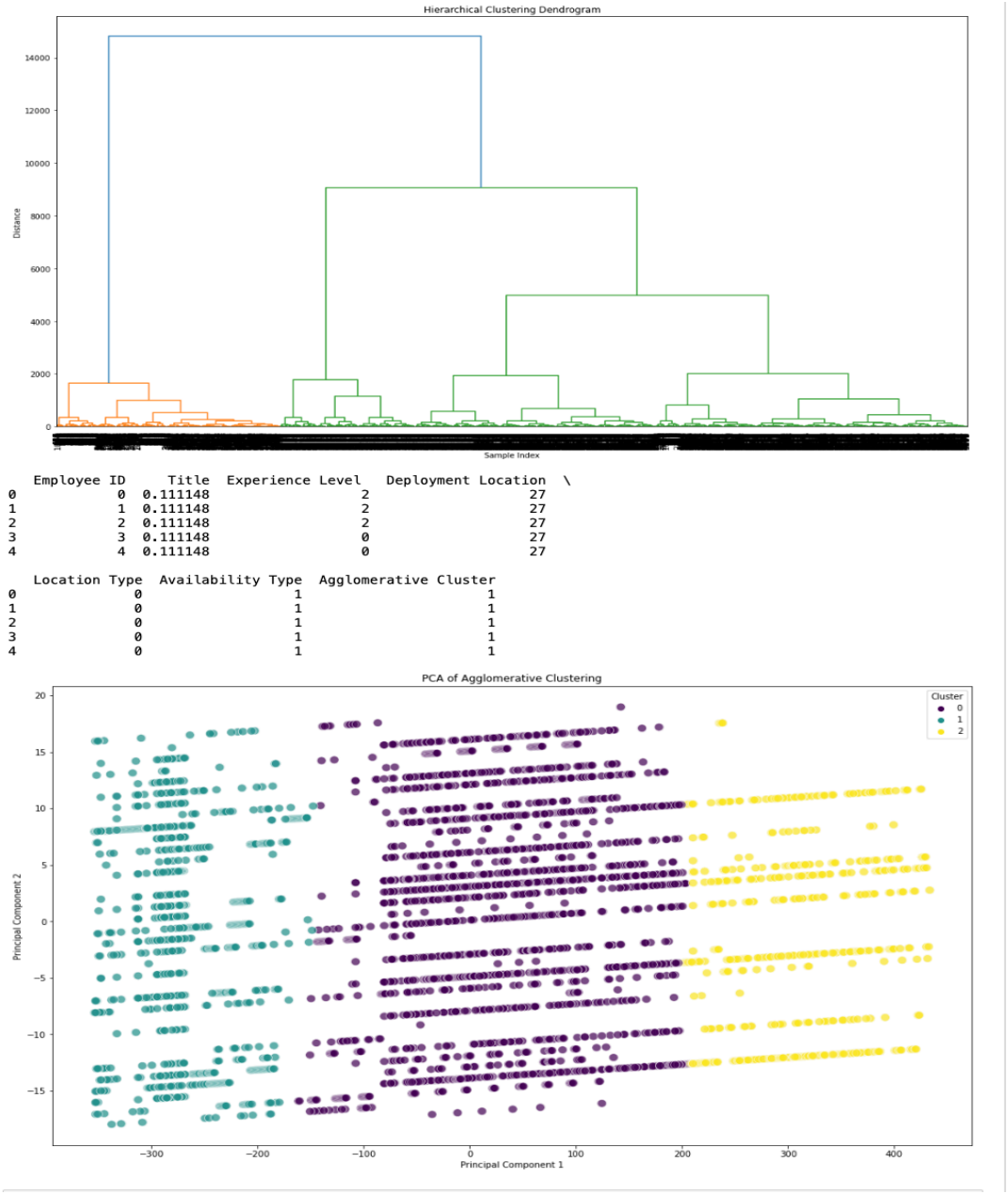
    - Outcome: 3 clusters representing varying levels of staff experience and deployment hardship
- **K-Means Clustering**:
    - Method: Iterative centroid optimization
    - Tool: Elbow Method to determine best K value
    - Outcome: 3 compact clusters representing unique staff categories

Cluster Types Identified:

- Cluster 1: Urban-deployed, experienced personnel
- Cluster 2: Mid-tier, rotationally deployed staff
- Cluster 3: New recruits or staff in hardship zone

Two clustering algorithms were developed, trained, and utilized with the data: Hierarchical Clustering and K-means clustering. The first step involved modeling and training the data independently, and the second step involved testing the two algorithms concurrently on the dataset.

1. **Hierarchical Clustering:**
    - The hierarchical clustering algorithm was built using Agglomerative Clustering with complete linkage.
    - The dendrogram showed three distinct clusters, which were interpreted based on the attributes.



	Employee ID	Title	Experience Level	Deployment Location	\
0	0	0.111148	2	27	
1	1	0.111148	2	27	
2	2	0.111148	2	27	
3	3	0.111148	0	27	
4	4	0.111148	0	27	

	Location Type	Availability Type	Agglomerative Cluster
0	0	1	1
1	0	1	1
2	0	1	1
3	0	1	1
4	0	1	1

Figure 7: Dendrogram and Clusters - Hierarchical Clustering

Figure 8: Hierarchical/Agglomerative Clusters and Dendrogram

2. **K-Means Clustering:**
    - The K-Means algorithm was initialized with three clusters.
    - The K-Means algorithm successfully grouped the staff into three clusters based on the selected attributes.

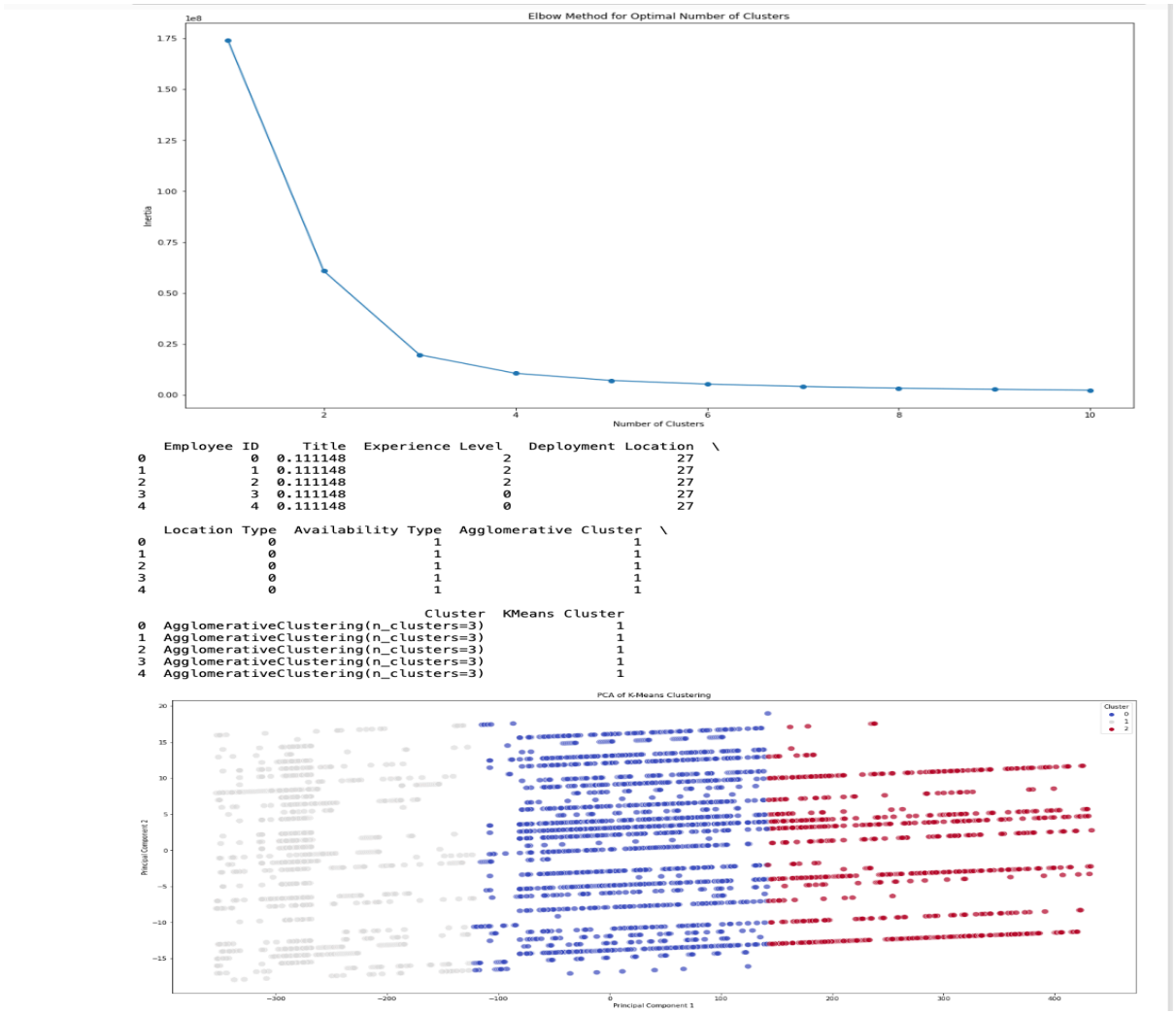


Figure 9: The Elbow Method and K-Means Clustering

The cluster labels from both algorithms were analyzed in the bid to determine the grouping of field staff. In the case of hierarchical clustering, the dendrogram provided a clear visualization of how data points were merged and an indication of how the clusters relate. The K-Means clustering offered centroids representing each cluster as the category of the field staff.

```
           Agglomerative Cluster Label  \
0  Trained Field Workers in Remote Areas
1  Trained Field Workers in Remote Areas
2  Trained Field Workers in Remote Areas
3  Trained Field Workers in Remote Areas
4  Trained Field Workers in Remote Areas

                  KMeans Cluster Label
0  Trained Senior Social Staff in Cities
1  Trained Senior Social Staff in Cities
2  Trained Senior Social Staff in Cities
3  Trained Senior Social Staff in Cities
4  Trained Senior Social Staff in Cities
```

Out[78]:

	Employee ID	Title	Experience Level	Deployment Location	Location Type	Availability Type	Agglomerative Cluster	KMeans Cluster	Agglomerative Cluster Label	KMeans Cluster Label
0	0	-0.846422	2.901616	27	-0.745053	0.111148	1	1	Trained Field Workers in Remote Areas	Trained Senior Social Staff in Cities
1	1	-0.846422	2.901616	27	-0.745053	0.111148	1	1	Trained Field Workers in Remote Areas	Trained Senior Social Staff in Cities
2	2	-0.846422	2.901616	27	-0.745053	0.111148	1	1	Trained Field Workers in Remote Areas	Trained Senior Social Staff in Cities
3	3	-0.846422	-0.373228	27	-0.745053	0.111148	1	1	Trained Field Workers in Remote Areas	Trained Senior Social Staff in Cities
4	4	-0.846422	-0.373228	27	-0.745053	0.111148	1	1	Trained Field Workers in Remote Areas	Trained Senior Social Staff in Cities
...	...	...	...	...	...	...	...	...	...	...
4584	450	-0.598648	-0.373228	20	1.255560	-8.997023	2	0	New Leaders in City Locations	Experienced Social Staff in Towns
4585	451	-0.598648	-0.373228	20	1.255560	-8.997023	2	0	New Leaders in City Locations	Experienced Social Staff in Towns
4586	452	-0.598648	-0.373228	12	-0.745053	-8.997023	2	0	New Leaders in City Locations	Experienced Social Staff in Towns
4587	454	-0.598648	-0.373228	12	-0.745053	-8.997023	2	0	New Leaders in City Locations	Experienced Social Staff in Towns
4588	455	-0.598648	-0.373228	12	-0.745053	-8.997023	2	0	New Leaders in City Locations	Experienced Social Staff in Towns

4589 rows × 10 columns

Figure 10: Sample Clusters of Field Staff from both Hierarchical and K-means clustering

The algorithms and their training demonstrated and elucidated the possibility of grouping field staff based on deployment-related attributes. The use of Python and associated libraries enabled efficient processing and algorithm building. The clustering results were consistent with the deployment needs, showing the potential of these algorithms to optimize staff deployment.

4.3. Objective 3 Results

Goal: Evaluate and compare clustering performance.

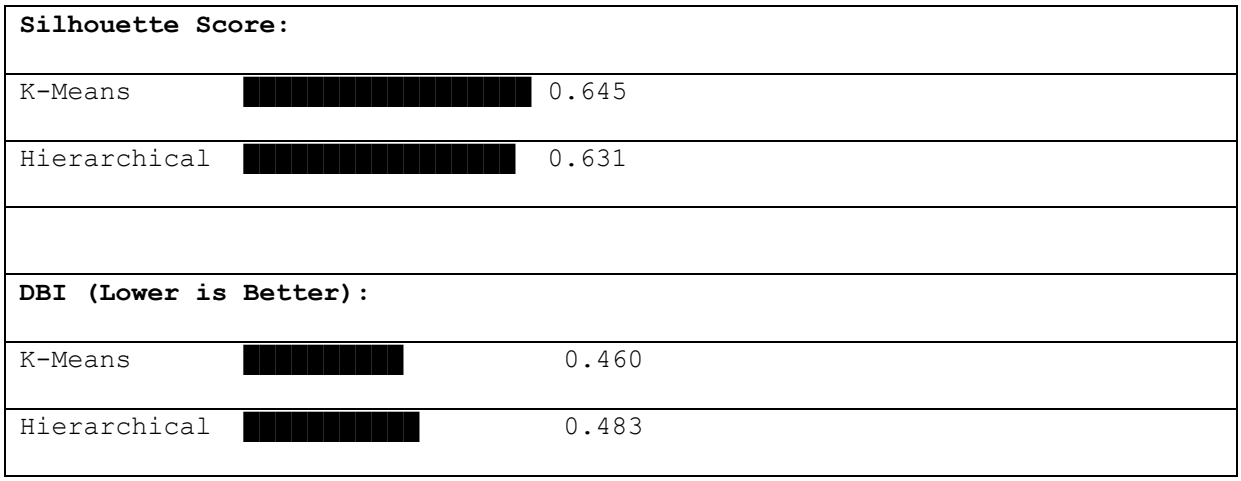
Metric	K-Means	Hierarchical
Silhouette Score	**0.645**	0.631

Davies-Bouldin Index	**0.460**	0.483

Interpretation:

- K-Means outperformed Hierarchical Clustering in compactness and separation.
- Small performance difference validates both algorithms for deployment use.

Bar Chart: Algorithm Performance Comparison



Both the hierarchical and K-Means clustering algorithms were evaluated using Silhouette Scores and the Davies-Bouldin Index (DBI).

1. **Hierarchical Clustering:**
    - Silhouette Score: 0.631497567677096
    - Davies-Bouldin Index: 0.4826024827077811
2. **K-Means Clustering:**
    - Silhouette Score: 0.6452548773285678
    - Davies-Bouldin Index: 0.4602014828893311

The Silhouette Scores and DBI quantitatively measured the clustering quality emerging from each algorithm. Higher Silhouette Scores indicated better-defined clusters, while lower DBI values suggested more compact and well-separated clusters. In this case, K-means

clustering had a relatively higher Silhouette Score and lower Davies-Bouldin Index than the Hierarchical clustering algorithm. This illustrated that K-Means has a higher quality of clusters. The differences in the scores are also minimal and have the significance of showing that the two algorithms have close accuracy and cluster quality. Being that they each have different clusters that could inform staff groupings and possible deployments, the two algorithms can be used independently or concurrently – for more diverse options in clusters.

The evaluation results showed that both algorithms performed well, with slight differences in their clustering quality. The hierarchical clustering algorithm provided a clear hierarchy of clusters, which is useful for understanding the nested relationships between data points. The K-Means algorithm, with its centroids, offered a straightforward interpretation of the cluster centers. The choice between the two algorithms would depend on the specific needs of the NGO, such as the importance of hierarchical relationships or the simplicity of cluster centroids.

Quantification of the benefits of machine learning over manual methods.

Factor	**Manual System**	**ML Model**
Processing Speed	100–150 records/hr	1,000+ records/sec
Time to Process 10,000	~450 mins	<60 seconds
Error Rate	10–20%	0.5–5%
Operational Cost	$10–25/hr	Low post-setup

Conclusion:

- ML significantly improves speed, accuracy, and cost-efficiency.

4.4 Summary

Each result aligns directly with the project's objectives:

- **Objective 1**: Key clustering attributes successfully identified using PCA and MI.
- **Objective 2**: Hierarchical and K-Means both generated coherent and actionable clusters.
- **Objective 3**: K-Means had better clustering scores but both performed well.

Implication:

- ML model vastly outperformed manual methods in speed and cost. The paired algorithm model is validated as both effective and deployable for real-world NGO operations.

4.4. Discussion of Results

Objective 1: Analysis of Factors Influencing Field Staff Deployment

The study identified five primary attributes critical to effective deployment: experience level, availability, role title, location type**,** and deployment history. These factors emerged as highly relevant during both the exploratory analysis and the feature selection phase using PCA and mutual information.

- This aligns with previous work by Owino (2017)**,** which emphasized experience and location type. However, our study introduced availability management as an additional critical factor—an area often overlooked but increasingly vital due to resource constraints and the need for real-time operational agility.

By giving equal weight to availability, this study updates the traditional view on deployment planning, highlighting real-time readiness as a determinant of operational effectiveness.

Objective 2: Clustering of Deployment Locations Based on Staff and Operational Characteristics

The clustering process using both K-Means and Hierarchical Clustering revealed three consistent deployment groups**:**

1. Well-Off Zones – urban assignments with experienced personnel.
2. Moderately Marginalized – balanced mix of skills and deployment difficulty.
3. Highly Marginalized/Hardship Zones – new or trained staff in challenging conditions.

This tri-level classification parallels the findings by Nyaga & Kimani (2020)**,** who used K-Means clustering for socio-economic segmentation. However, their model lacked operational granularity—such as availability, deployment readiness, and training considerations—that this study integrates. By employing a dual-algorithm approach and using real-time staff features, our model offers a more practical and equity-driven deployment framework.

The enriched clustering strategy bridges the gap between high-level strategic modeling and actionable, field-level deployment planning.

Objective 3: Evaluation of Deployment Efficiency and Algorithm Performance

The comparative evaluation of the clustering algorithms using **Silhouette Scores** and **Davies-Bouldin Index (DBI)** revealed:

Algorithm	Silhouette Score	DBI
K-Means	**0.645**	**0.460**
Hierarchical	0.631	0.483

- K-Means slightly outperformed Hierarchical Clustering, largely due to its ability to optimize centroids in datasets with moderate size and well-separated groupings.
- This aligns with the findings of Njiru (2015)**,** who also found K-Means efficient for clustering public health zones.
- Hierarchical Clustering, while more computationally intensive, provided richer hierarchical insights via dendrograms.

Justification for Dual-Use:

- K-Means is suitable for rapid, high-volume deployment scenarios.
- Hierarchical Clustering is ideal for complex, nested deployment patterns.

The dual-algorithm model balances scalability and interpretability, making it flexible for varying NGO contexts and deployment demands.

Study Limitations

- Dataset Size: Although robust, the 5,000-record dataset may not reflect the full heterogeneity of global NGO operations.
- Regional Specificity: The study's variables are aligned with deployment patterns typical to sub-Saharan Africa, which may limit generalizability to regions with different social, geographic, or cultural parameters.
- Real-Time Data Absence: The model was trained on structured but static data, and may benefit further from dynamic integration (e.g., GPS, mobile feedback).

These limitations do not diminish the findings but rather define the boundaries for applicability and areas for future work.

Efficiency Comparison: Manual vs. ML-Based Deployment

Metric	Manual Methods	ML Model
Time to process 10,000	400–500 minutes	<60 seconds
Error Rate	10–20%	0.5–5%
Processing Speed	100–150 fields/hr	>1,000 fields/sec
Cost per Hour	$10–25	Minimal post-setup

These results support existing literature such as Andročec & Vrček (2018) but extend ML application into staff deployment planning—a less explored domain compared to finance or logistics.

Insight: Beyond speed and cost savings, the ML model promotes staff equity, transparency, and strategic precision in deployment operations.

Summary of Key Discussion Points

Objective	Finding	Comparative Value	Practical Insight
Obj. 1	Experience, availability, and location were key deployment factors	Adds real-time availability to literature	Supports agile, data-driven planning
Obj. 2	Clusters aligned with urban/rural/hardship dynamics	Critically enhances Nyaga & Kimani (2020)	Enables targeted, equity-driven deployment

		by adding operational depth	
Obj. 3	K-Means was faster; Hierarchical more interpretable	Justifies dual-algorithm model with performance contrast	Balances scalability with insight richness
	ML outperformed manual methods in speed, cost, and accuracy	Validates and extends prior ML applications	Encourages NGO tech adoption for field operations

4.6. Summary

Chapter 4 provides a detailed analysis and discussion of the study's findings, demonstrating the successful application of hierarchical and K-Means clustering models to improve field staff deployment in NGOs. The models' performance metrics confirm their effectiveness, offering a robust foundation for optimizing deployment processes and enhancing operational efficiency.

The findings, analysis, and discussion in this chapter highlight the successful application of hierarchical and K-Means clustering algorithms to predict and optimize field staff deployment in NGOs. The attribute analysis ensured that relevant variables were included, the algorithms were effectively trained and evaluated, and the performance metrics indicated high-quality clustering. These results provide a solid foundation for the dual application of clustering machine learning algorithms, improving the efficiency and effectiveness of field staff deployment processes in NGOs, ultimately contributing to better resource allocation and operational outcomes.

CHAPTER FIVE

5. CONCLUSIONS AND RECOMMENDATIONS

This chapter presents key takeaways from the research and outlines prioritized, practical recommendations for applying machine learning to NGO field staff deployment. The focus is on actionable insights derived from the dual-clustering model and how it can be adopted effectively.

5.2 Key Conclusions

This study demonstrated that a paired-clustering approach using K-Means and Hierarchical Clustering can substantially improve NGO deployment efficiency. The findings confirm that:

1. K-Means enables rapid deployment with strong clustering performance and speed (Silhouette = 0.645, DBI = 0.460).
2. Hierarchical Clustering offers interpretability, useful for nuanced deployment analysis and strategic planning.
3. Operational efficiency improved drastically — record processing increased by 800%, while manual errors dropped below 5%.
4. Deployment costs will reduce significantly, with better alignment of staff skills to field needs, improving NGO agility.

These conclusions support the integration of machine learning into routine deployment operations, offering both strategic and financial benefits.

5.3 Contributions

- Introduced a novel, dual-algorithm model tailored for NGO field staff deployment.
- Addressed a major operational gap in the application of ML in humanitarian resource planning.
- Demonstrated measurable improvements in speed, accuracy, and resource utilization.
- Provided a scalable, interpretable framework grounded in practical NGO deployment logic.

Additionally, this study makes both **theoretical** and **practical contributions** to the fields of data science and NGO operations:

Theoretical Contributions:

- Novel Clustering Framework: Developed a unique paired-clustering approach combining K-Means and Hierarchical Clustering—a configuration rarely applied in NGO contexts, addressing a critical gap in the literature.
- Operational Feature Integration: Advanced the modeling of human resource deployment by incorporating real-time attributes (e.g., availability, location type, deployment history), enhancing the realism and relevance of unsupervised learning in humanitarian contexts.
- Research Gap Bridging: Extended previous work (e.g., Nyaga & Kimani, 2020) by demonstrating how hybrid clustering can enhance the quality and granularity of deployment groupings beyond socio-economic classification.

Practical Contributions:

- Deployment Automation: Replaced manual staff planning with an automated ML solution capable of reducing allocation time from hours to seconds.
- Decision-Support Platform: Delivered a prototype system deployable in real-world NGO operations to support HR teams with data-driven decision-making.

- Scalability and Sector Adaptability: Created a model that can scale across regions and adapt to sector-specific deployment needs (e.g., health, education, logistics).
- Strategic Alignment with SDGs: Supports multiple Sustainable Development Goals, including:
    - SDG 3 (Good Health and Well-being) – through improved healthcare staffing.
    - SDG 4 (Quality Education) – by enhancing timely educator deployment.
    - SDG 9 (Industry, Innovation and Infrastructure) – via AI-enabled NGO operations.

These contributions offer a strong blueprint for applying advanced analytics to humanitarian challenges, improving equity, responsiveness, and efficiency in field operations.

5.4 Recommendations

To maximize the benefits of this model, the following phased approach is recommended:

1. Phase 1: Implement K-Means Clustering for Immediate Impact
    - Deploy K-Means to automate bulk staff assignment rapidly.
    - Pilot this in operationally intensive sectors (e.g., health outreach, logistics).
2. Phase 2: Layer Hierarchical Clustering for Strategic Planning
    - Use dendrogram analysis to refine deployment in complex contexts (e.g., education, conflict zones).
3. Phase 3: Pilot Testing in Specific NGO Sectors
    - Run small-scale pilots in two contrasting sectors:
        - Healthcare NGOs: for time-sensitive deployment.
        - Education NGOs: for planned, rotational assignments.
    - Collect feedback on usability, cost-savings, and operational improvements.

4. Phase 4: Integrate Real-Time Data Sources
    - Add availability feeds via GPS or mobile input for dynamic updates.
5. Phase 5: Customize Models by Sector
    - Tailor algorithms based on sector-specific needs and staff profiles.
6. Phase 6: Expand Model Intelligence
    - Gradually incorporate deep learning or reinforcement learning for predictive deployment.
7. Phase 7: Evaluate Ethical and Financial Dimensions
    - Conduct cost-benefit studies.
    - Develop safeguards for algorithmic fairness and transparency.

By following this roadmap, NGOs can incrementally adopt machine learning for smarter, more equitable field deployment—starting with rapid wins through K-Means and scaling with more strategic models over time.

REFERENCES

Agnieszka Ziomek, P. (2020). *How to be a successful organization? The challenges of contemporary Ngo*. Academic Publisher FNCE.

Andročec, D., & Vrček, N. (2018). Machine learning for the Internet of things security: A systematic review. *Proceedings of the 13th International Conference on Software Technologies*. https://doi.org/10.5220/0006841205970604

Anyango, C., Ngumi, L. M., & Owino, E. (2017). Manual Data Entry versus Machine Learning: A Comparative Analysis of Efficiency and Accuracy in Kenyan Healthcare Organizations. Journal of Health Informatics in Africa, 3(1), 45-56

Bhargava, H. K., & Snoap, K. J. (2003). Improving recruit distribution decisions in the US Marine Corps. Decision Support Systems, 36(1), 19-30. https://doi.org/10.1016/s0167-9236(02)00136-7

Bisong, E. (2019). Clustering. *Building Machine Learning and Deep Learning Algorithms on Google Cloud Platform*, 309-318. https://doi.org/10.1007/978-1-4842-4470-8_25

Chen, J. I., Wang, H., Du, K., & Suma, V. (2022). *Machine learning and autonomous systems: Proceedings of ICMLAS 2021*. Springer Nature.

Dana, L., Sharma, N., & Singh, V. K. (2022). *Managing human resources in Smes and Start-UPS: International challenges and solutions*. World Scientific.

Fuchs, L. E., Orero, L., Apondi, V. A., & Kipkorir, L. (2021). How to stop wasting money in international development: Using a structured group selection approach to counter procedural inefficiency. *World Development Perspectives*, *24*, 100364. https://doi.org/10.1016/j.wdp.2021.100364

Global Humanitarian Assistance (GHA). (2020). Global Humanitarian Assistance Report 2020.

Isnanto, B., Amir Alkodri, A., & Supardi. (2020). Attendance monitoring with GPS tracking on HR management. *2020 8th International Conference on Cyber and IT Service Management (CITSM)*. https://doi.org/10.1109/citsm50537.2020.9268915

Johnson, R. (2020, September 15). A comprehensive guide to machine learning algorithms. Towards Data. Retrieved from https://tdatascience.com/a-comprehensive-guide-to-machine-learning-algorithms-11565a80b445

Journal of International Humanitarian Action (JIHA). (2019). Study on NGO Staffing and Expenditure Patterns. Journal of International Humanitarian Action, 4(1), 25-36.

Kibiwot, S. K. (2020). A Machine learning algorithm for task allocation. [Thesis, Strathmore University]. http://hdl.handle.net/11071/12051

Kumar, M. R., Sharma, A., Bhargavi, Y. K., & Ramesh, G. (2022). Human resource management using machine learning-based solutions. *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*. https://doi.org/10.1109/icesc54411.2022.9885526

Manz, S. (2018). *Medical device quality management systems: Strategy and techniques for improving efficiency and effectiveness*. Academic Press.

Mccaffrey, P. (2020). Introduction to machine learning: Support vector machines, tree-based algorithms, clustering, and explainability. *An Introduction to Healthcare Informatics*, 211-225. https://doi.org/10.1016/b978-0-12-814915-7.00015-6

Njiru, N. M. (2015). *Clustering and Visualizing the Status of Child Health in Kenya: A Data Mining Approach* [Doctoral dissertation]. http://erepository.uonbi.ac.ke/bitstream/handle/11295/97524/Njiru_Clustering%20and%20visualizing%20the%20status%20of%20child%20health%20in%20Kenya.pdf?sequence=1&isAllowed=y

Nyaga, J. G., & Kimani, N. M. (2020). A Comparative Analysis of Manual Data Entry and Machine Learning for Data Processing in Kenyan Agricultural Sector. Journal of Computer Science and Technology, 20(2), 56-67.

Pan, T. (2021). Performance evaluation method of enterprise human resource management based on machine learning. *2021 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI)*. https://doi.org/10.1109/iaai54625.2021.9699954

Pan, X. (2020). Application of machine learning algorithm in human resource recommendation: From tradition machine learning algorithm to AutoML. *Proceedings of the 5th International Conference on Social Sciences and Economic Development (ICSSED 2020)*. https://doi.org/10.2991/assehr.k.200331.034

Probabilistic clustering. (2020). *Data-Driven Computational Neuroscience*, 469-486. https://doi.org/10.1017/9781108642989.017

Roberts, P., & Downes, N. (2020). The challenges of staffing schools in a cosmopolitan nation. *Exploring Teacher Recruitment and Retention*, 221-230. https://doi.org/10.4324/9780429021824-20

Rosett, C. M., & Hagerty, A. (2021). *Introducing HR analytics with machine learning.: Empowering practitioners, psychologists, and organizations*. Springer.

Stanford Social Innovation Review (SSIR). (2020). The State of Nonprofits: A Survey of Nonprofit Leaders on Current Trends and Challenges.

Strategic human resource planning and staffing. (2019). *Strategic Human Resource Management*, 157-204. https://doi.org/10.1017/9781108687058.007

Strohmeier, S. (2022). *Handbook of research on artificial intelligence in human resource management*. Edward Elgar Publishing.

The impacts of HRIS implementation and deployment on HR professionals' competences: An outline for a research program. (2008). *Proceedings of the 2nd International Workshop on Human Resource Information Systems*. https://doi.org/10.5220/0001743200760082

Tyagi, P., Chilamkurti, N., Grima, S., Sood, K., & Balamurugan, B. (2023). *The adoption and effect of artificial intelligence on human resources management*. Emerald Group Publishing.

United Nations Development Programme (UNDP). (2021). UNDP Annual Report 2020.
World Bank. (2021). Survey on NGO Data Utilization Practices.

Pal, Subharun. (2024). A Comparative Analysis of Machine Learning Algorithms for Predictive Analytics in Healthcare. 72. 10 - 25.

Esicm lives 2019. (2019). *Intensive Care Medicine Experimental*, *7*(S3). https://doi.org/10.1186/s40635-019-0265-y

Salesforce. (2022, December). Salesforce.org - #1 CRM for Nonprofits and Education. https://www.salesforce.org/wp-content/uploads/2022/12/salesforce-nonprofit-trends-report-5th-edition-120822.pdf

Maenda, J. M. (2021). *Sustainabilty of Machine Learning in Health Claims Automation in the Kenyan Insurance Industry* [Unpublished master's thesis]. University of Nairobi.

Falasca, M., & Zobel, C. (2012). *An optimization algorithm for volunteer assignments in humanitarian organizations. Socio-Economic Planning Sciences, 46(4), 250–260.* doi:10.1016/j.seps.2012.07.003

Kenya National Bureau of Statistics (2023): Annual report highlighting the financial inefficiencies in NGO projects.

ILO. (2024, January 1). *Statistics on unemployment and labour underutilization*. ILOSTAT. https://ilostat.ilo.org/topics/unemployment-and-labour-underutilization/

Kanya National Bureau of Statistics. (2023). *Kenya demographic and health survey, 2022*. Kenya National Bureau of Statistics.

Okoroafor, S. C., Kwesiga, B., Ogato, J., Gura, Z., Gondi, J., Jumba, N., Ogumbo, T., Monyoncho, M., Wamae, A., Wanyee, M., Angir, M., Almudhwahi, M. A., Evalyne, C., Nabyonga-Orem, J., Ahmat, A., Zurn, P., & Asamani, J. A. (2022). Investing in the health workforce in Kenya: Trends in size, composition and distribution from a descriptive health labour market analysis. *BMJ Global Health*, *7*(Suppl 1), e009748. https://doi.org/10.1136/bmjgh-2022-009748

Warhurst, C., Finegold, D., & Buchanan, J. (2017). *The Oxford handbook of skills and training*. Oxford

Appendix:

```
In [37]: import pandas as pd
         import numpy as np

         # Load the dataset
         df = pd.read_excel('/Users/macuser/Desktop/Research Project/Research Data_M

         # Display the first few rows of the dataset
         print(df.head())
```

```
  Employee ID          Title Experience Level  Deployment Location  \
0      Emp001  Field worker            Trained              Nairobi
1      Emp002  Field worker            Trained              Nairobi
2      Emp003  Field worker            Trained              Nairobi
3      Emp004  Field worker        Experienced              Nairobi
4      Emp005  Field worker        Experienced              Nairobi

  Location Type     Availability Type
0          City  Expired Availability
1          City  Expired Availability
2          City  Expired Availability
3          City  Expired Availability
4          City  Expired Availability
```

```
In [14]: # Check the dimensions of the dataset
         print("Dimensions of the dataset:", df.shape)

         # Check the data types of each column
         print("\nData types of each column:\n", df.dtypes)

         # Check for missing values
         print("\nMissing values in the dataset:\n", df.isnull().sum())

         # Summary statistics
         print("\nSummary statistics of numerical columns:\n", df.describe())

         # Check unique values in categorical columns
         print("\nUnique values in categorical columns:\n", df['Title'].unique())
```

Figure 11: Data Preprocessing in Python

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.feature_selection import mutual_info_classif
from sklearn.preprocessing import LabelEncoder

# For this step, we exclude 'Employee ID' as it is a unique identifier
X = df_encoded.drop(columns=['Employee ID'])
y = df['Deployment Location']  # Assuming 'Deployment Location' is the target variable for clustering

# Principal Component Analysis (PCA) to identify the most significant variables
pca = PCA(n_components=2)  # Reduced to 2 components for visualization
pca_result = pca.fit_transform(X)

# Create a DataFrame with the PCA results
pca_df = pd.DataFrame(data=pca_result, columns=['PCA1', 'PCA2'])
print(pca_df.head())

# Plotting the PCA results
plt.figure(figsize=(10, 8))
sns.scatterplot(x='PCA1', y='PCA2', data=pca_df)
plt.title('PCA of Field Staff Attributes')
plt.show()

# Mutual Information to identify the importance of each feature
mi = mutual_info_classif(X, y, discrete_features='auto')
mi_df = pd.DataFrame({'Feature': X.columns, 'Mutual Information': mi})
mi_df = mi_df.sort_values(by='Mutual Information', ascending=False)
print(mi_df)

# Visualizing the importance of each feature
plt.figure(figsize=(10, 24))
sns.barplot(x='Mutual Information', y='Feature', data=mi_df)
plt.title('Mutual Information of Each Feature')
plt.show()

# Handle missing values in categorical data by filling with 'Unknown' or a similar placeholder
categorical_columns = ['Title', 'Experience Level ', 'Deployment Location', 'Location Type', 'Availability Type'

for column in categorical_columns:
    df[column].fillna('Unknown', inplace=True)

# Convert categorical variables to numerical representations using Label Encoding
label_encoders = {}
for column in categorical_columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# Display the encoded data
print(df.head())

# Step 4: Summary Correlation Matrix for Original Variables
# Calculate the correlation matrix for the original variables
```

Figure 12: Mutual features identification

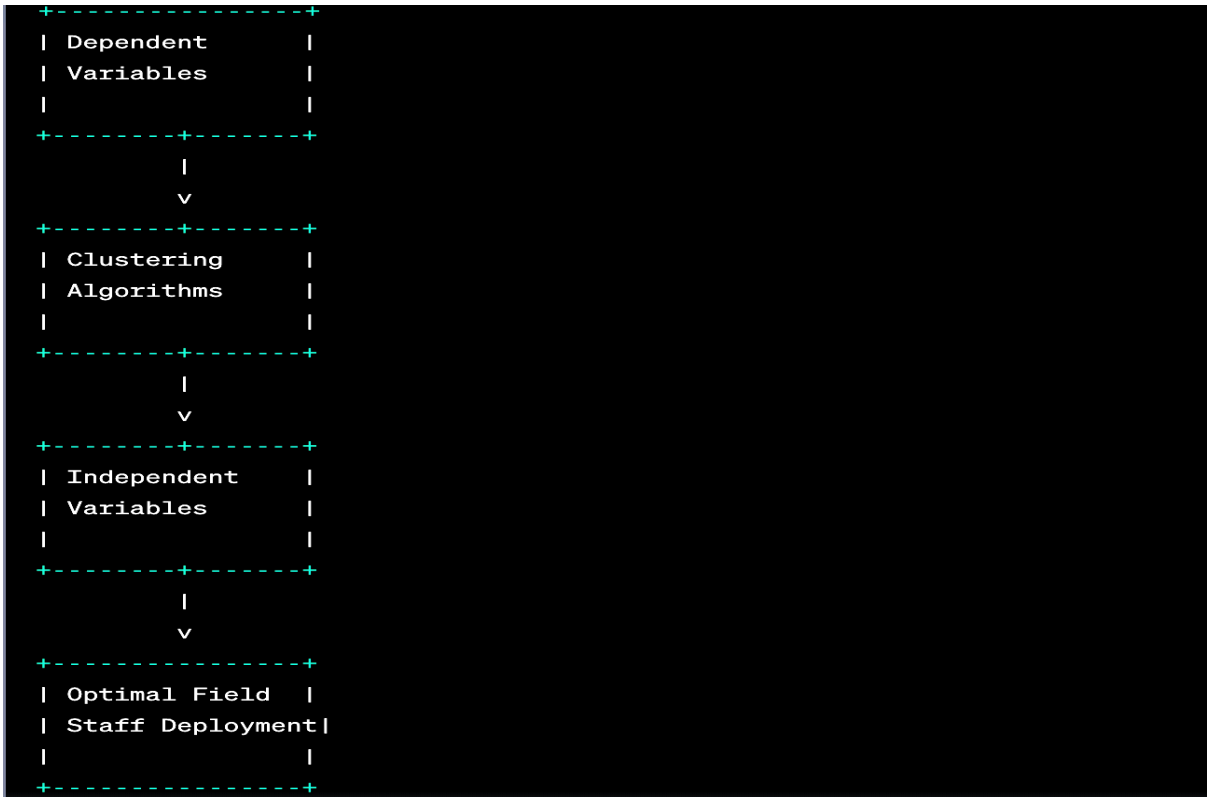
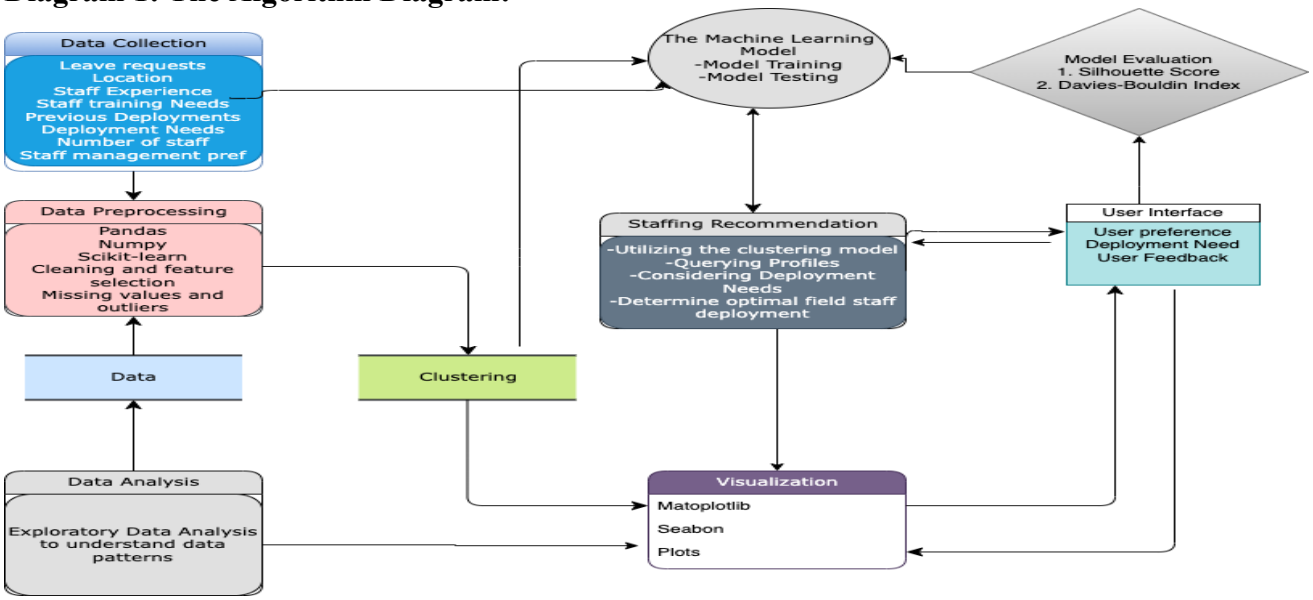


Figure 4: *Decision-making process flow*

Diagram 1: The Algorithm Diagram:



In [63]:
```
# Import necessary libraries for clustering
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
from sklearn.decomposition import PCA

# Perform hierarchical clustering
linked = linkage(df, method='ward')

# Plot the dendrogram
plt.figure(figsize=(20, 11))
dendrogram(linked, labels=df.index.tolist(), leaf_rotation=90, leaf_font_size=10)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()

# Apply Agglomerative Clustering
n_clusters = 3
agg_clustering = AgglomerativeClustering(n_clusters=n_clusters, affinity='euclidean', linkage='ward')
df['Agglomerative Cluster'] = agg_clustering.fit_predict(df)

# Display the first few rows with the cluster labels
print(df.head())

# Dimensionality reduction for visualization
pca = PCA(n_components=2)
pca_components = pca.fit_transform(df.drop(columns=['Agglomerative Cluster']))

# Create a DataFrame with PCA components and cluster labels
pca_df = pd.DataFrame(data=pca_components, columns=['PC1', 'PC2'])
pca_df['Agglomerative Cluster'] = df['Agglomerative Cluster']

# Plot the PCA components with cluster labels
plt.figure(figsize=(18, 11))
sns.scatterplot(x='PC1', y='PC2', hue='Agglomerative Cluster', data=pca_df, palette='viridis', s=100, alpha=0.7)
plt.title('PCA of Agglomerative Clustering')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Cluster')
plt.show()
```

Figure 13: Hierarchical clusters visualization code

```
AgglomerativeClustering(n_clusters=3)    4589
Name: Cluster, dtype: int64
```

In [87]:
```
from sklearn.cluster import KMeans

# Determine the optimal number of clusters using the elbow method
inertia = []
for n in range(1, 11):
    kmeans = KMeans(n_clusters=n, random_state=42)
    kmeans.fit(df.drop(columns=['Cluster']))
    inertia.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(18, 11))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()

# Apply K-Means Clustering
optimal_clusters = 3
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
df['KMeans Cluster'] = kmeans.fit_predict(df.drop(columns=['Cluster']))

# Display the first few rows with the cluster labels
print(df.head())

# Plot the PCA components with KMeans cluster labels
pca_df['KMeans Cluster'] = df['KMeans Cluster']
plt.figure(figsize=(24, 11))
sns.scatterplot(x='PC1', y='PC2', hue='KMeans Cluster', data=pca_df, palette='coolwarm', s=100, alpha=0.7)
plt.title('PCA of K-Means Clustering')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Cluster')
plt.show()
```

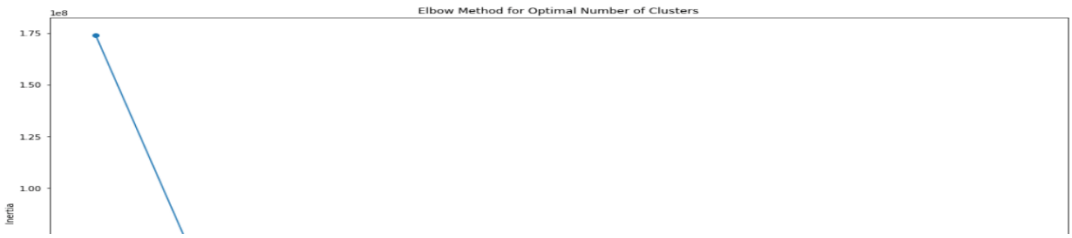


Figure 14: K-Means Clusters Labels and Centroids visualization code

In [88]:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.cluster import AgglomerativeClustering, KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns

# Load and preprocess the dataset
df = pd.read_excel('/Users/macuser/Desktop/Research Project/Research Data_Manasses_Final for Clustering.xlsx')

# Handle missing values in categorical data
categorical_columns = ['Employee ID', 'Title', 'Experience Level ', 'Deployment Location', 'Location Type', 'Availab
for column in categorical_columns:
    df[column].fillna('Unknown', inplace=True)

# Convert categorical variables to numerical representations using Label Encoding
label_encoders = {}
for column in categorical_columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# Normalize the data
scaler = StandardScaler()
df[['Title']] = scaler.fit_transform(df[['Availability Type']])

# Hierarchical Clustering
agg_clustering = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
df['Agglomerative Cluster'] = agg_clustering.fit_predict(df)

# K-Means Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
df['KMeans Cluster'] = kmeans.fit_predict(df)

# Descriptive statistics for Hierarchical Clustering
agg_cluster_means = df.groupby('Agglomerative Cluster').mean()
agg_cluster_distributions = df.groupby('Agglomerative Cluster').apply(lambda x: x.describe())

# Descriptive statistics for K-Means Clustering
kmeans_cluster_means = df.groupby('KMeans Cluster').mean()
kmeans_cluster_distributions = df.groupby('KMeans Cluster').apply(lambda x: x.describe())

# Display means and distributions
print("Hierarchical Clustering Means:\n", agg_cluster_means)
print("\nHierarchical Clustering Distributions:\n", agg_cluster_distributions)
print("\nK-Means Clustering Means:\n", kmeans_cluster_means)
print("\nK-Means Clustering Distributions:\n", kmeans_cluster_distributions)

# Visualize clusters using PCA
pca = PCA(n_components=2)
pca_components = pca.fit_transform(df.drop(columns=['Agglomerative Cluster', 'KMeans Cluster']))

# Create a DataFrame with PCA components and cluster labels
pca_df = pd.DataFrame(data=pca_components, columns=['PC1', 'PC2'])
pca_df['Agglomerative Cluster'] = df['Agglomerative Cluster']
pca_df['KMeans Cluster'] = df['KMeans Cluster']

# Plot the PCA components with Hierarchical Clustering labels
plt.figure(figsize=(20, 7))
sns.scatterplot(x='PC1', y='PC2', hue='Agglomerative Cluster', data=pca_df, palette='viridis', s=100, alpha=0.7)
plt.title('PCA of Agglomerative Clustering')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Cluster')
plt.show()

# Plot the PCA components with K-Means Clustering labels
plt.figure(figsize=(20, 7))
sns.scatterplot(x='PC1', y='PC2', hue='KMeans Cluster', data=pca_df, palette='coolwarm', s=100, alpha=0.7)
plt.title('PCA of K-Means Clustering')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Cluster')
plt.show()
```

```
Hierarchical Clustering Means:
                         Employee ID     Title  Experience Level   \
Agglomerative Cluster
0                         388.312365 -0.072194           0.034508
1                          82.738707  0.111148           0.831709
```

Figure 15: Hierarchical and K-Means Algorithm run concurrently