Thesis submitted to the

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA, ALGERIA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

in Partial Fulfillment of the Requirements for the Degree of:

Master's in Computer Science

Specialization: Informatique decisionnel et optimisation

By

CHAIMA, BOUZIDI

NESSRINE KHAWLA, ABDELLAOUI

Entitled

A Comparative Analysis of Density Based Clustering Algorithms in Complex Datasets

Under the supervision of

LOUNNAS BILAL

Jury Members

MOKHTARI RABAH	University of M'sila	President
LOUNNAS BILAL	University of M'sila	Reporter
CHALABI BAYA	University of M'sila	Examiner

June, 2025

الملخص

تُجري هذه الأطروحة تحليلاً مقارناً لخوارزميات التجميع القائمة على الكثافة، مع التركيز على أدائها في مجموعات البيانات المعقدة. وتتناول الدراسة خوارزميات شائعة مثل DBSCAN وOPTICS وHDBSCAN، وتُقيّمها وفقاً لمعايير متنوعة، تشمل جودة التجميع، والكفاءة الحسابية، ومتانة البيانات أمام الضوضاء. ومن خلال التجارب على مجموعات البيانات الواقعية والتركيبية، يهدف البحث إلى تحديد نقاط القوة والضعف لكل خوارزمية، مما يُقدم رؤى قيّمة لاختيار الأساليب المناسبة في مختلف تطبيقات استخراج البيانات.

الكلمات المفتاحية: التجميع المعتمد على الكثافة، DBSCAN، OPTICS، HDBSCAN جودة التجمعات، الكفاءة الحسابية، مقاومة الضوضاء، تحليل مقارن، تنقيب البيانات، مجموعات البيانات الحقيقية، مجموعات البيانات الاصطناعية.

Abstract

This dissertation conducts a comparative analysis of density-based clustering algorithms, focusing on their performance in complex datasets. The study examines popular algorithms such as DBSCAN, OPTICS, and HDBSCAN, evaluating them across diverse criteria, including cluster quality, computational efficiency, and robustness to noise. Through experiments on real-world and synthetic datasets, the research aims to identify the strengths and limitations of each algorithm, providing valuable insights for selecting appropriate methods in various data mining applications.

Keywords: Density-Based Clustering, DBSCAN, OPTICS, HDBSCAN, Cluster Quality, Computational Efficiency, Noise Robustness, Data Mining, Comparative Analysis, Synthetic Datasets, Real-World Data.

Dedication

"To the first person who taught me love and sacrifice, To the one who stayed up countless nights for me and prayed for me endlessly, To my beloved mother… the source of tenderness and the warmth of my life, All my words and every achievement I make are the fruit of your patience and hard work.

To the one who taught me the meaning of manhood, honesty, and dedication, To my dear father… my first support and steady guide, You deserve all my respect and appreciation; you are the compass that led me to the right path.

To my two beloved sisters, The companions of my soul, the light of my days, and my support through sorrow and joy, You have always been my refuge and the reason behind my smiles—thank you for being you.

And to my two dear brothers, My childhood partners and the constant presence in every challenge and moment of laughter, You have always been there with your humor, protection, and beautiful spirits. Thank you for being an inseparable part of my journey.

To my whole family, you are the heartbeat of my life, Every step I take toward success is thanks to you, I proudly dedicate this work to you, Hoping it will bring joy to your hearts just as you've brought joy to mine"

Abdellaoui nessrine khawla

Dedication

“To my mother and father... You are the truest meaning of love and prayer, the unwavering light when the roads grew dim. I would not have reached this achievement without your patience, your prayers, and the peace your presence brings to my heart. To you, every moment of pride and every word written in these pages is lovingly dedicated.

To my sister, who resembles me more than I resemble myself, And to my dear brothers, You were the light in my darkest days and the strength when I needed it most. May God bless you always, and may you remain close to my heart as you have always been.

To my nephews, You are the flowers of my days and the joy of my heart. I watched you grow, and with you, my dreams grew too. I hope this work becomes a source of inspiration for you and a doorway to your own successes.

And to all those who carried love for me in their hearts, who believed in me silently, who were a source of warmth or strength — whether they realized it or not, this work carries something”

Bouzidi chaima

Acknowledgements

All praise is due to Allah, the Lord of the Worlds, and peace and blessings be upon the most honorable of messengers.

We extend our sincere gratitude and profound appreciation to everyone who contributed to the completion of this humble work. We especially thank Professor **Bilal Lounnas**, who graciously supervised this dissertation, providing invaluable support, insightful guidance, and sound advice that illuminated our research path. To him, we offer our utmost respect and appreciation.

Furthermore, we express our deep gratitude and appreciation to our esteemed professors in the Department of Computer Science at the Faculty of Mathematics and Computer Science, University of M'Sila, for the knowledge and wisdom they imparted throughout our years of study. May Allah reward them abundantly.

In this context, we also wish to extend our heartfelt thanks and recognition to our dear family, our pillar of support and aid, for their continuous encouragement and boundless patience, which served as a constant motivator for us to move forward. To our friends (My best friend, Radja Imane), who were excellent companions on this academic journey, we thank them for every moment of support and assistance. We pray to Allah Almighty that this work is sincerely for His sake and that it may be beneficial.

Contents

List of Tables

List of Figures

Introduction

The field of data science and intelligent systems has witnessed unprecedented growth in recent years, emerging as a critical pillar for informed decision-making across scientific, industrial, and commercial sectors. As the complexity, scale, and heterogeneity of data continue to expand, the demand for robust, scalable, and intelligent frameworks to process and interpret data becomes increasingly urgent. This work explores essential pillars of modern data-centric systems by investigating the foundations of Data Science, the role of Artificial Intelligence (AI) and Machine Learning (ML), and the practical implementation of clustering algorithms for knowledge discovery and pattern recognition.

- **Key research aspects:**
  - Current integration of Data Science, AI, and ML in decision-support systems
  - Theoretical and practical aspects of Data Mining and Knowledge Discovery
  - Comparative evaluation of clustering algorithms on complex datasets
- **Core problems identified:**
  - Lack of unified frameworks for extracting actionable knowledge from large-scale data
  - Challenges in handling noisy, high-dimensional, or unstructured data
  - Difficulties in algorithm selection based on data properties and performance metrics
- **Study objectives:**
  1. Establish a clear conceptual foundation linking Data Science, AI, ML, and KDD
  2. Explore key data mining processes with a focus on clustering as a knowledge discovery technique
  3. Conduct an experimental evaluation of clustering algorithms, including density-based and benchmark methods
- **Methodological framework:**
  - Conceptual exploration of data-driven intelligent systems
  - In-depth study of clustering techniques, including DBSCAN, HDBSCAN, DPC, K-means, and GBC

  - Experimental testing using real-world and synthetic datasets with quantitative performance evaluation

- **Document organization:**

  - **Machine learning and data minig** (Conceptual foundations of Data Science, AI, ML, KDD, and Data Warehousing
  - **Clustering** (Data mining processes and clustering techniques
  - **Comparison and results** (Experimental comparison of clustering algorithms with evaluation metrics

Chapter 1: Machine learning and data mining

1.1 Introduction

In the age of information, data has become a cornerstone for strategic decision-making across all sectors. As the volume, variety, and velocity of data continue to grow, the need for advanced tools and methodologies to extract meaningful insights has become increasingly critical. Within this context, Data Science has emerged as an interdisciplinary field that combines statistics, programming, and data analysis to uncover patterns and generate actionable knowledge. Closely related to this field are Artificial Intelligence (AI)—which seeks to emulate human cognitive capabilities—and Machine Learning (ML), a subfield of AI that enables systems to learn from data and improve their performance autonomously. To support analysis and intelligent learning processes, two additional concepts are essential: Knowledge Discovery in Databases (KDD), which provides a systematic framework for extracting useful knowledge from large data sets, and Data Warehousing, which facilitates the organized storage and integration of data from various sources, enabling efficient retrieval and analysis. This chapter explores these core concepts in detail, highlighting their interconnections and emphasizing their collective role in modern data-driven systems across a wide range of applications such as business intelligence, healthcare, scientific research, and digital marketing.

1.2 Data science

Data science is an interdisciplinary field focused on extracting meaningful knowledge and insights from data by integrating principles and techniques from statistics, computer science, and domain-specific knowledge. It involves a comprehensive workflow that includes data collection, cleaning, transformation, analysis, and communication of results to support informed decision-making. Beyond the use of algorithms and analytical tools, data science requires a deep understanding of the context in which data is applied, along with critical thinking and creativity. It plays a central role in various modern applications such as targeted marketing, financial forecasting, customer relationship management, and fraud detection. The field emphasizes a structured, iterative process that blends theory with practice, and encourages hands-on learning through real-world problem solving. [1] [2]

1.2.1 The interdisciplinary nature

All these fields coming together are where Data Science is defined as Math and Stats give analytical and mathematics model of real-time problem and Business where these things can be used for strategic decision making. With a nexus at the meetings of machine learning, software development, and research, Data Science is a field through which it is possible to extract insights from data, and leverage those insights in different industries.

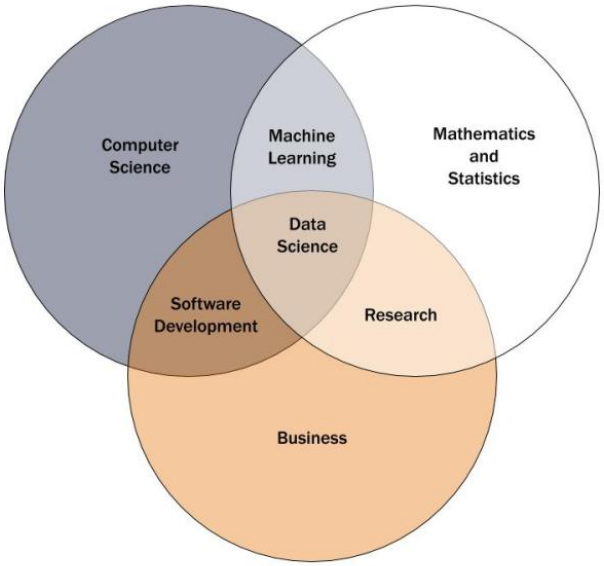


Figure 1.1: The interdisciplinary nature of data science

[3]

1.2.2 Key components of data science

Data science is a multifaceted discipline that integrates several critical components to transform raw data into meaningful insights. These components work together to support data-driven decision-making and predictive modeling.

Data

At the foundation of data science lies data itself. High-quality, relevant, and well-sourced data—whether structured or unstructured—is essential for effective analysis. Data can be collected from diverse sources such as sensors, social media, images, documents, and human activity.

Data visualization

Data visualization involves presenting complex datasets in graphical or pictorial formats to enhance interpretability, especially for non-technical audiences. Common tools include charts, graphs, and maps. Effective visualization helps reveal patterns, trends, and insights that might remain hidden in raw data.

Statistics and probability

Statistics and probability are fundamental to data analysis and machine learning. They enable the collection, organization, interpretation, and presentation of data, and provide the mathematical foundation for making inferences and estimating uncertainties. These tools are crucial for understanding data distributions, relationships, and predictive modeling.

Programming

Programming languages like Python and R are essential for managing, analyzing, and modeling data. Python is widely used for its flexibility and extensive libraries, while R excels in statistical computing and data visualization. Proficiency in at least one of these languages is necessary for implementing data science workflows.

Machine learning

Machine Learning (ML) is the engine of predictive analytics in data science. It involves training algorithms to detect patterns and make predictions or decisions without being explicitly programmed. ML is widely applied in areas such as fraud detection, customer segmentation, and recommendation systems.

Data engineering

Data engineering focuses on the architecture, tools, and systems used to collect, process, and store data efficiently. It ensures the reliability and scalability of data pipelines, enabling seamless access and analysis. Data engineers build the infrastructure upon which data science depends.

Domain expertise

Domain expertise refers to a deep understanding of the specific field in which data science is applied—be it finance, healthcare, marketing, etc. This contextual knowledge is vital for interpreting results accurately and building models that reflect real-world dynamics. Domain experts guide the framing of questions and the relevance of outcomes.[4]

1.3 Machine learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that empowers machines to automatically learn from data and past experiences without being explicitly programmed for each specific task. Through the identification of hidden patterns within large datasets, ML algorithms are capable of making accurate predictions, adapting to new data, and improving their performance over time with minimal human intervention. Rather than relying on predefined models or equations, ML leverages computational algorithms that iteratively learn from data to extract meaningful insights. This adaptive learning process enables systems to operate autonomously, growing and evolving as they are exposed to more data. Machine learning plays a vital role across a wide range of real-world applications, including image and speech recognition, natural language processing, recommendation systems, fraud detection, financial portfolio optimization, and task automation. It is also at the core of advanced technologies such as autonomous vehicles, robotics, and drones, where dynamic learning and real-time adaptation are crucial. A prominent subfield of ML is Deep Learning, which mimics human cognitive processes by learning from layered representations of data. It often outperforms traditional ML techniques in tasks requiring high-level abstraction and large-scale data processing.[5][6]

1.3.1 Categories of machine learning

Machine Learning is typically divided into several categories:

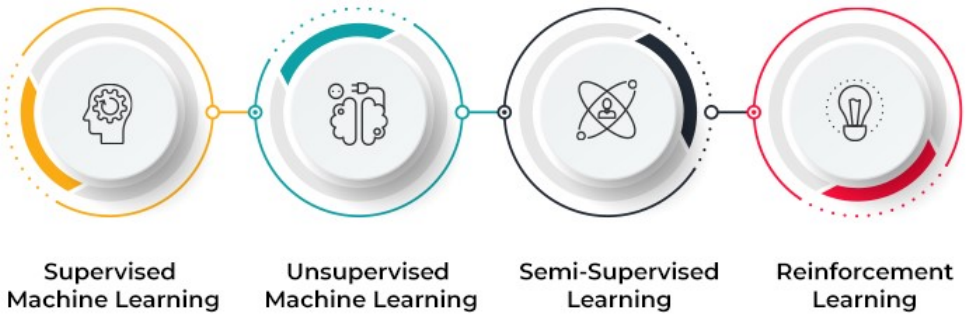


Figure 1.2: Categories of Machine Learning

[7]

Supervised Learning

Supervised learning relies on labeled data to train a model. The goal is to establish a relationship between an input variable and an output variable. It is divided into two main categories: classification, which predicts categorical outputs (e.g., detecting whether an email is spam or

not), and regression, which predicts continuous values (e.g., forecasting temperature). Popular algorithms include logistic regression, decision trees, and support vector machines (SVM).

Unsupervised learning

Unlike supervised learning, this approach works without labeled data. The algorithm analyzes the data and identifies underlying structures or patterns. It is divided into two main types: clustering, which groups similar data points (e.g., customer segmentation), and association, which identifies relationships between variables (e.g., product recommendations on e-commerce websites). Common algorithms include K-Means, DBSCAN, and the Apriori algorithm.

Semi-supervised learning

This approach combines the benefits of both supervised and unsupervised learning by using a small set of labeled data alongside a larger set of unlabeled data. This improves model performance while reducing the cost of labeling data. A real-world example is a student learning a concept in class under a teacher's guidance and then reviewing the material independently.

Reinforcement learning

Reinforcement learning is based on a reward and penalty system that allows an agent to learn through trial and error. The goal is to optimize the agent's decisions by maximizing rewards. There are two types of reinforcement: positive reinforcement, which encourages desirable behaviors by providing rewards, and negative reinforcement, which encourages avoiding undesirable actions. This technique is widely used in fields such as video games, robotics, and finance.

1.3.2 Machine learning work flow

The typical workflow for developing and deploying machine learning models includes the following steps: [8]

- Problem definition: Clearly articulate the problem to be solved and determine how machine learning can provide a solution.
- Data collection: Gather relevant data from various sources, ensuring it is representative of the problem domain.
- Data preprocessing: Clean and transform the data to handle missing values remove inconsistencies and prepare it for analysis.

- Feature selection: Choose appropriate algorithms and models that are well-suited to the problem and data characteristics.
- Training: Use the prepared data to train the model allowing it to learn from the patterns present.
- Evaluation: Assess the model's performance using metrics relevant to the problem, such as accuracy precision recall or F1-score.
- Hyperparameter tuning: Optimize the model's hyperparameters to improve performance and prevent overfitting.
- Deployment: Integrate the trained model into a production environment where it can make predictions on new data.
- Monitoring and maintenance: Continuously monitor the model's performance in the real world and update it as necessary to maintain accuracy and relevance.

1.4 Artificial intelligence

Artificial intelligence refers to computer systems that can perform tasks typically associated with human intelligence, such as making predictions, recognizing objects, understanding speech, and generating natural language. These systems learn by processing large volumes of data and identifying patterns that guide their decision-making. Often, humans supervise the learning process by reinforcing correct decisions and discouraging incorrect ones. However, some AI systems are designed to learn independently, without human supervision. Over time, AI systems become more effective at performing specific tasks, enabling them to adapt to new inputs and make decisions without being explicitly programmed. In essence, artificial intelligence is about teaching machines to think and learn like humans, with the aim of automating tasks and solving problems more efficiently.[9] [10]

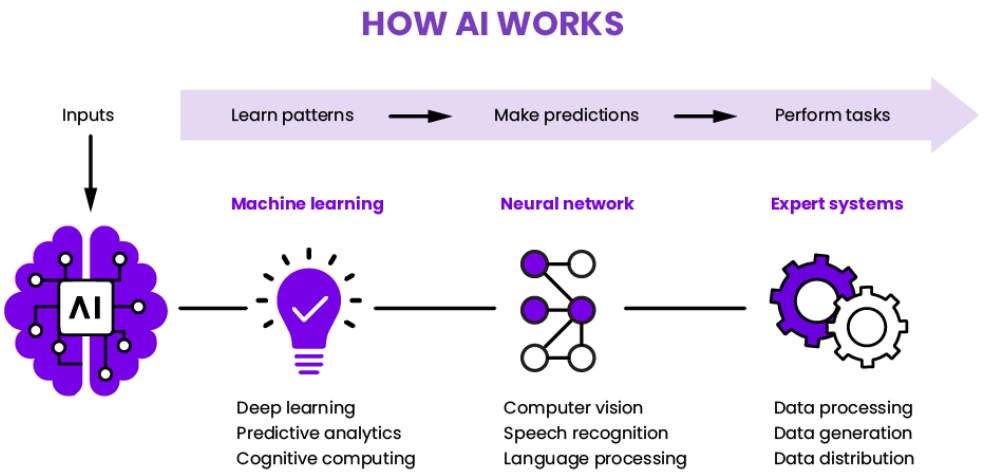


Figure 1.3: How Artificial Intelligence Operates

[11]

1.4.1 Branches of artificial intelligence

Artificial intelligence encompasses seven key branches, each playing a crucial role in advancing intelligent systems. First, computer vision enables machines to "see" and interpret images and videos by using convolutional neural networks to classify images, track objects, and recognize faces, such as in facial unlock on smartphones. Second, fuzzy logic mimics human reasoning when dealing with uncertainty by evaluating degrees of truth rather than binary yes/no answers; it's applied in systems like smart car braking. Third, expert systems are programs designed to emulate human expertise in specific domains by applying logical inference rules to solve complex problems, for example, cadet assists doctors in early cancer detection. Fourth, robotics involves machines capable of performing automated tasks, often enhanced with AI for human interaction and complex actions, like the humanoid robot Sophia. Fifth, machine learning allows systems to learn from data and improve performance without explicit programming, with types including supervised, unsupervised, and reinforcement learning, used in prediction, classification, and pattern detection. Sixth, neural networks and deep learning, inspired by the human brain, consist of layers of artificial neurons that progressively learn from data, powering applications such as speech recognition and machine translation. Finally, natural language processing (NLP) enables computers to understand and process human language in text and speech, applied in virtual assistants, chatbots, sentiment analysis, and spam detection. Together, these branches empower AI systems to become increasingly adaptive and intelligent in meeting human needs.[12]

1.4.2 Technologies in artificial intelligence

Artificial Intelligence (AI) is powered by a variety of advanced technologies that enable machines to perform tasks traditionally requiring human intelligence. These technologies allow AI systems to learn, process information, and make decisions. From understanding language to recognizing patterns in images, these AI technologies are transforming industries and shaping the future of automation. [13]

- Natural language generation (NLG): A subfield of AI that converts structured data into human-like text, enabling systems to communicate insights and ideas in a clear and coherent manner. It's widely used in customer service, automated reporting, and content creation.
- Speech recognition: This technology allows machines to capture and interpret spoken language by converting human speech into readable and processable text. It powers voice assistants, call center automation, and real-time transcription tools.
- Machine learning platforms: Comprehensive environments that provide algorithms, APIs, data handling tools, and training systems to build intelligent applications that learn from data. These platforms are widely used for predictive modeling, classification, and decision-making.
- Virtual agents: AI-powered programs that simulate conversations with humans, often through text or voice. They are used in chatbots, virtual assistants, and smart home interfaces to provide automated yet human-like interactions.
- Decision management: This AI application integrates logic and business rules into systems to automate and enhance decision-making processes. It's used to build intelligent systems that support real-time, data-driven business decisions.
- AI Optimized hardware: Specialized hardware—such as AI chips and GPUs—designed specifically to accelerate and efficiently process AI tasks, including training complex neural networks and running deep learning models.
- Deep learning platforms: Advanced machine learning systems that use neural networks mimicking the human brain to process large datasets and detect intricate patterns. Applications include speech recognition, image analysis, and autonomous systems.
- Robotic process automation (RPA): Technology that automates repetitive and rule-based tasks traditionally performed by humans, such as data entry and invoice processing, thereby improving accuracy and operational efficiency.

- Natural language processing (NLP): A field focused on enabling computers to understand, interpret, and generate human language. NLP powers tools like sentiment analysis, machine translation, and intelligent assistants.
- Biometrics: AI technologies that identify and analyze human physical and behavioral characteristics—such as fingerprints, facial features, and voice—for authentication and interaction, commonly used in security and user experience enhancement.
- Cyber defense: AI-driven systems designed to detect, prevent, and respond to cyber threats. They analyze user behavior and network activity using machine learning to identify anomalies and protect digital infrastructure.
- Content creation: AI tools that automatically generate written or multimedia content, including news articles, marketing copy, and social media posts. Tools like Wordsmith use NLG to produce data-driven narratives at scale.
- Emotion recognition: AI systems that detect and interpret human emotions through facial expressions, speech tones, or physiological signals. This technology is applied in security, customer service, and market research.
- Image recognition: An AI process that involves identifying and classifying objects, people, text, or actions within images or videos. It's used in facial recognition, medical imaging, and automated visual inspections.
- Marketing automation: The use of AI to streamline and optimize marketing processes such as customer segmentation, campaign management, and personalized content delivery—leading to more efficient and targeted marketing strategies.

1.4.3 Challenges in artificial intelligence

Artificial intelligence (AI) has revolutionized industries, from healthcare and finance to automation and customer service. However, despite its rapid advancements, AI faces several challenges that hinder its full potential. These challenges can be broadly categorized into technical and non-technical issues:

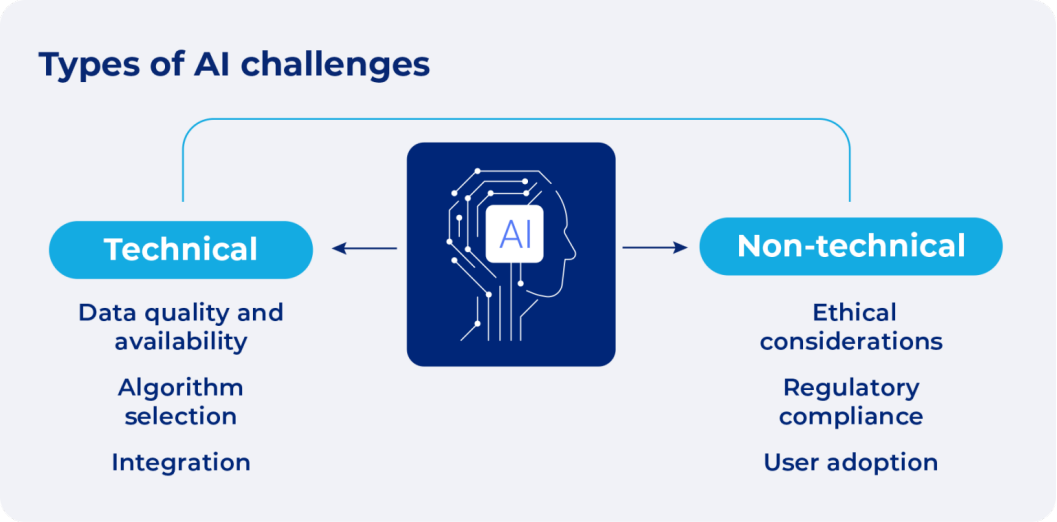


Figure 1.4: types of AI challenges

[14]

Technical challenges

- Data quality and availability : Poor data quality hinders the successful deployment of AI and ML projects. 1 Even the most advanced AI algorithms can yield flawed results if the underlying data is of low quality. We explain the importance of data quality in AI, the challenges organizations encounter, and the best practices for ensuring high-quality data.
- Algorithm selection : Choosing the right algorithm is crucial for AI success. A poorly chosen algorithm can lead to inefficient or inaccurate results.
- Integration : Implementing AI into existing systems can be complex, requiring adjustments to ensure compatibility and optimal performance.

No-Technical challenges

- Ethical considerations : AI raises ethical concerns, particularly regarding bias and discrimination. If training data contains prejudices, AI models can reinforce them.
- Regulatory compliance : Companies must navigate complex legal frameworks to ensure that their AI applications '
- User adoption : AI adoption can be hindered by a lack of trust or understanding among users, limiting its effectiveness.

1.4.4 Applications of artificial intelligence

Artificial intelligence is an effective tool used in various fields, most notably in... [15]

E-commerce

Artificial Intelligence (AI) has revolutionized the e-commerce industry by enhancing customers' shopping experiences and optimizing businesses' operations. AI-powered recommendation engines analyze customer behavior and preferences to suggest products, leading to increased sales and customer satisfaction. Additionally, AI-driven chat-bots provide instant customer support, resolving queries and guiding shoppers through their purchasing journey. **Example** Amazon uses AI to recommend products to its users based on their browsing history, past purchases, and preferences. This personalization boosts engagement and sales by showing customers items they are more likely to buy.

Education

The next AI application is it's use in the betterment of education! AI in education is transforming how students learn and how educators teach. Adaptive learning platforms use AI to customize educational content based on each student's strengths and weaknesses, ensuring a personalized learning experience. AI can also automate administrative tasks, allowing educators to focus more on teaching and less on paperwork. **Example** Platforms like Simplilearn use AI algorithms to offer course recommendations and provide personalized feedback to students, enhancing their learning experience and outcomes.
AI apps are revolutionizing user experiences across various domains. Some of these top AI apps include:

Chat-gpt

Chat-GPT is an advanced language model developed by Open AI that excels in generating human-like text responses. Its key feature is the ability to understand and respond to a wide range of queries, making it ideal for applications such as customer support, content creation, and interactive conversations

Google gemini

Another common AI app is Gemini. Google Gemini integrates cutting-edge AI to deliver highly personalized search results and recommendations. Its key feature is the ability to analyze user behavior and preferences to provide tailored content and suggestions, enhancing the overall search and browsing experience.

Amazon alexa

Amazon Alexa is a versatile voice assistant designed to control smart home devices, answer questions, and perform various tasks through voice commands. Its key feature is its extensive compatibility with a wide range of smart devices and services, making everyday tasks more convenient and hands-free.

Elsa speak

ELSA Speak is an AI-powered app focused on improving English pronunciation and fluency. Its key feature is the use of advanced speech recognition technology to provide instant feedback and personalized lessons,

Google maps

One of the most used and popular AI apps is Maps. Google Maps is a comprehensive navigation app that uses AI to offer real-time traffic updates and route planning. Its key feature is the ability to provide accurate directions, traffic conditions, and estimated travel times, making it an essential tool for travelers and commuters.

Snap-chat

Most common between the younger generation and marketers is the next AI app: Snapchat! Snap-chat incorporates artificial intelligence to enhance its popular AR filters and photo editing features. Its key feature is the use of AI-driven image processing to create engaging and interactive visual content, making it a favorite among social media users.

Starry-ai

The next on the list of top AI apps is Starry-AI, an innovative app that uses artificial intelligence to generate stunning artwork based on user inputs. Its key feature is the ability to create unique and visually appealing art pieces, showcasing the creative potential of AI and providing users with personalized digital art experiences.

1.4.5 Ethics of artificial intelligence

Ethics is a system of principles and values that helps individuals and societies distinguish between right and wrong, guiding human behavior in accordance with concepts such as justice, responsibility, and respect. These principles are rooted in religious, cultural, philosophical, and social foundations, and they form the basis for laws, social norms, and interpersonal relationships. AI ethics, on the other hand, is a multidisciplinary field that addresses the ethical

challenges arising from the development and deployment of artificial intelligence technologies. Its main goal is to maximize the positive impact of AI in areas like healthcare, education, and the economy, while minimizing potential risks such as algorithmic bias, privacy violations, lack of transparency, and diminished accountability. This field also raises critical questions related to fairness, freedom, and responsibility—such as: Who is accountable for decisions made by AI systems? How can we ensure that these systems do not perpetuate discrimination or harm vulnerable groups? AI ethics seeks to ensure that technological progress is guided by shared human values, protecting human dignity and rights, and promoting the fair and safe use of modern technologies. [16]

1.5 Knowledge discovery in databases (KDD)

Knowledge discovery in databases (KDD) is defined as a systematic and non-trivial process aimed at extracting valuable information and knowledge from large datasets. This process involves several stages, starting with data preparation and cleaning, passing through the application of data mining techniques, and ending with the interpretation and evaluation of the results, with the goal of uncovering hidden patterns that support decision-making or provide a deeper understanding of the studied phenomena. [17] [18] [19] [20] [21] [22]

1.5.1 What is the kdd process?

The Knowledge Discovery in Databases (KDD) process is a systematic approach aimed at extracting meaningful patterns and useful knowledge from large volumes of data. This process consists of several interrelated stages, as outlined below :

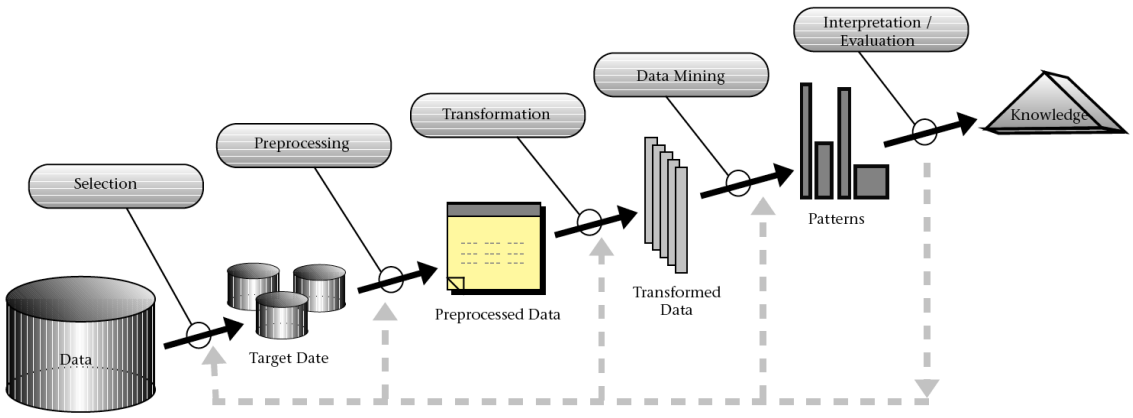


Figure 1.5: The technical sequence of knowledge discovery in databases (kdd) processes

Understanding the domain

In this initial phase, the overall objective of the analysis is clarified, including whether the problem is descriptive or predictive. A thorough understanding of the data context and the application domain is essential for guiding the subsequent steps.

Data selection

Relevant data is selected from databases or other data sources. This step involves identifying appropriate attributes that relate to the analytical objective and ensuring that the dataset is sufficient in volume and quality.

Data preprocessing

This critical stage focuses on cleaning the data by handling missing values, duplicates, inconsistencies, and errors. The goal is to prepare high-quality, standardized data that is suitable for further analysis.

Data transformation

The selected and cleaned data is transformed into a format appropriate for data mining. This involves dimensionality reduction, normalization, aggregation, and discretization, aiming to improve data efficiency and relevance.

Data mining

This is the core stage of the KDD process, where the selected algorithms are applied to the transformed data to discover meaningful patterns, trends, relationships, and groupings through techniques such as classification, clustering, association rule mining, and regression.

() **The terms "knowledge discovery" and "data mining" represent distinct concepts Knowledge Discovery in Data (KDD)** refers to the overall process of extracting valuable knowledge from data. This process involves evaluating and possibly interpreting the extracted patterns to determine which patterns qualify as actionable knowledge. The KDD process also includes selecting appropriate encoding schemes, data preprocessing, sampling techniques, and transforming the data to prepare it for the next step, which is data mining. On the other hand, Data Mining refers to the application of algorithms to directly extract patterns from the data, without the preliminary steps involved in the KDD process.

Pattern evaluation

The discovered patterns are evaluated in terms of their validity, usefulness, and relevance to the original objectives. Irrelevant or low-quality patterns are discarded at this stage.

Knowledge representation

The extracted knowledge is presented in a clear and comprehensible format, such as reports, visualizations, or dashboards, to facilitate interpretation by end users.

Knowledge deployment

Finally, the discovered knowledge is applied to real-world scenarios, such as decision-making processes or system improvements, to enhance organizational effectiveness. [17] [23] [24]

1.5.2 Fundamental concepts and terminologies in the knowledge discovery in data (KDD) methodology

Term	Definition
Data	A set of facts, F.
Pattern	An expression E in a language L describing facts in a subset F' of F.
Process	KDD is a *multi-step process* involving data preparation, pattern searching, knowledge evaluation, and refinement with iteration after modification.
Valid	Discovered patterns should be true on new data with some degree of certainty. Generalize to the future (other data).
Novel	Patterns must be novel (should not be previously known).
Useful	Actionable; patterns should potentially lead to some useful actions.
Understandable	The process should lead to human insight. Patterns must be made understandable in order to facilitate a better understanding of the underlying data.

Table 1.1: Key terms and definitions in the kdd process

[25]

1.5.3 Applications of kdd

Data mining techniques are extensively utilized across multiple sectors, including banking, finance, and healthcare. In this context, five exemplary applications have been selected to illustrate the fundamental strengths and practical value of data mining.

1. Healthcare and insurance: In the healthcare sector, KDD techniques help identify best medical practices based on data analysis, leading to cost reduction and improved patient outcomes. They are also used to predict patient conditions across various categories, allowing critical care to be delivered more efficiently and at the right time. Additionally, KDD can be applied to clinical data to discover effective medical treatments, analyze patient behavior to forecast future visits to healthcare facilities, and detect fraudulent activities by healthcare providers.
   In the insurance sector, KDD is used to analyze customer behavior and predict which clients are likely to purchase new policies, identify high-risk customer patterns, and detect fraudulent actions. In the pharmaceutical industry, data analysis can be used to evaluate the performance of sales representatives and determine the most effective marketing activities in the upcoming months.
   Examples of kdd applications in the healthcare and insurance sectors :

   (a) Predicting patient readmission to hospitals (Healthcare): Hospitals often face the challenge of patient readmission within a short period after discharge, which results in increased costs and additional strain on the healthcare system. By applying Knowledge Discovery in Databases (KDD) techniques, historical patient data—such as diagnoses, treatment types, length of stay, and demographic information—can be analyzed. Using classification algorithms, such as decision trees or logistic regression, predictive models can be developed to identify patients who are at high risk of readmission. This allows medical teams to intervene early, provide personalized care, and improve treatment outcomes.

   (b) Detecting fraud in health insurance (Insurance) : Fraudulent claims represent a major challenge for insurance companies. KDD techniques are utilized to detect such cases by analyzing large volumes of claim data. Methods such as clustering and anomaly detection are applied to identify unusual patterns in the data—for example, frequent claims within a short time span or transactions that appear suspicious. This analysis enables early detection of fraudulent behavior, reducing financial losses and enhancing the efficiency of insurance operations.

2. Market basket analysis and its applications : Market Basket Analysis is a key data mining technique used primarily in the retail and marketing sectors to analyze customer purchasing behavior. It involves studying transaction data to discover patterns of frequently purchased items together. This analysis helps companies better understand customer preferences, enabling them to design more effective promotional strategies, such as targeted deals, discounts, and product placements. Beyond retail, data mining techniques are also widely applied in sales and marketing to enhance customer service, identify cross-selling opportunities, and increase direct marketing effectiveness.

Additionally, customer retention can be improved by identifying patterns that indicate potential customer churn, allowing for proactive engagement. Furthermore, risk assessment and fraud detection rely on data mining to uncover unusual or inappropriate behaviors, supporting better decision-making and loss prevention.
Example: A telecommunications company analyzes customer usage data—such as call frequency, subscription duration, and customer complaints. Through this analysis, the company identifies patterns that indicate a customer may soon cancel their subscription, such as a decline in call activity or repeated complaints. Based on these insights, the company proactively offers special deals or additional services to retain the customer. **Result:** Reduced customer churn rates and increased customer loyalty.

3. **Education :** Data mining plays a significant role in the education sector through a specialized method known as Educational Data Mining (EDM). EDM generates valuable patterns that can be utilized by both learners and educators to enhance the learning experience. By applying EDM techniques, several educational tasks can be performed, including:

   - Predicting student admissions to higher education.
   - Profiling students based on their learning behaviors.
   - Forecasting student performance and placement opportunities.
   - Assessing teachers' teaching performance
   - Supporting curriculum development.

   Although the application of data mining in education is still evolving, it aims to create methodologies for extracting knowledge from educational data. These techniques are designed to serve a variety of purposes, such as researching how educational support impacts students, helping students plan their future learning needs, and advancing the science of learning itself. Educational institutions can use these insights not only to predict student performance in exams but also to make informed decisions about instructional strategies, ultimately improving the quality of education. **Example:** A university analyzes historical academic records using data mining to predict which first-year students are at risk of failing specific courses. Based on the results, the university offers additional tutoring sessions and academic support to those students. **Result:** Improved student performance and reduced dropout rates.

4. **the Financial and banking sector:** The financial and banking sector has witnessed a rapid increase in data generation due to digitalization, creating vast repositories of customer and transaction information. Data mining techniques play a critical role in helping financial institutions analyze this large volume of data to uncover meaningful

patterns, correlations, and trends that support strategic decision-making. For example, credit card companies can analyze customer transaction histories to identify individuals most likely to respond to new financial products. Similarly, banks can apply data mining methods to detect credit card fraud, determine spending patterns across different customer groups, and identify loyal or high-value clients. Moreover, data mining facilitates the extraction of relevant customer information that supports personalized services and targeted marketing.

By leveraging these insights, managers in the financial and banking sectors can improve customer acquisition, enhance retention strategies, and deliver more effective financial services.

Example: By using data mining techniques, a bank detected an unusual pattern in a customer's credit card usage—multiple purchases from different countries within a short period. This behavior was flagged as potentially fraudulent, prompting the bank to immediately freeze the card and notify the customer.

Result: Early fraud detection and reduced financial losses.

5. **Transportation** In the transportation sector, companies with diverse services and a large sales force can leverage data mining techniques to identify the most promising prospects for their offerings. For instance, a major consumer goods organization can apply data mining to enhance its supply chain processes and optimize its business cycle with retailers.

   Data mining can be used to:

   (a) Determine the most efficient distribution schedules for outlets.

   (b) Analyze loading patterns to improve logistics and operational efficiency.

   Example: A large transportation company uses data mining techniques to analyze data from its previous shipments. By analyzing loading patterns and the schedules at which shipments occur, the company can optimize distribution schedules and determine the best times to deliver goods to stores.

   Result: Improved operational efficiency and reduced logistical costs. [26] [27]

6. **Cybersecurity:** Cybersecurity is a branch of information security that focuses on protecting computer systems, networks, and sensitive data from unauthorized access, theft, damage, and service disruptions. Its importance has increased with the widespread use of the Internet, wireless networks, and smart devices like smartphones and iot technologies.

   As cyber threats continuously evolve, organizations must adopt dynamic security measures, combining technical tools and employee training to stay ahead of potential risks. Key applications of cybersecurity include data protection, network and cloud security,

application security, raising user awareness, regulatory compliance, and securing critical infrastructures. Common security tools include firewalls, intrusion detection systems, and antivirus software, all aimed at maintaining a safe digital environment.
ExampleA financial company implements cybersecurity measures like firewalls and intrusion detection systems to protect its online banking platform. Additionally, it regularly trains its employees on recognizing phishing emails to prevent data breaches.
ResultEnhanced protection of customer data, reduced risk of cyberattacks, and increased trust in the company's services. [28] [29]

7. **Social media and web:**Social media are Internet-based platforms that enable users to communicate, interact, and share user-generated content. In today's digital era, communication has evolved significantly, with social media platforms like Facebook, Instagram, linkedin, pinterest, flickr, tumblr, and twitter dominating the online space, not only for social interaction but also for commerce and business activities.
   Data mining in social media involves analyzing and extracting patterns, correlations, and trends from raw social media data. It allows for drawing conclusions about user behaviors, preferences, and online activities. Just as mineral mining extracts valuable elements from raw ore, social media mining filters large volumes of user-generated data to identify meaningful insights related to user behavior, content sharing, social connections, and purchasing habits.
   These insights are highly valuable for businesses, governments, and non-profit organizations, helping them to design strategies, launch new products or services, and enhance decision-making processes. Additionally, it supports targeted advertising and academic research by providing a deeper understanding of digital interactions and trends.
   Example: A digital marketing company analyzes Instagram user data, such as the types of posts users interact with and the hashtags they frequently use. Through this analysis, the company discovers that a certain audience segment is highly interested in fitness products. As a result, the company tailors its advertising campaigns to target this audience with customized sports-related advertisements.[30][31]

1.5.4 Challenges in kdd

Some of the main current challenges in the research and application of KDD are summarized below:

1. **Large databases :**The growth of databases containing hundreds of fields and tables, millions of records, and sizes reaching multiple gigabytes has become commonplace, with terabyte-sized (10^{12} bytes) databases now emerging. Managing such large volumes of data presents a significant challenge in KDD. Addressing this requires the

development and use of more efficient algorithms, data sampling techniques, approximation methods, and massively parallel processing systems to ensure effective and scalable data analysis.

2. **Missing and noisy data:** The presence of missing and noisy data remains a major obstacle in KDD, especially within business databases. For instance, some fields in U.S. census data exhibit error rates as high as 20% . Additionally, key attributes may be absent when databases are not originally designed with data discovery purposes in mind. To overcome this challenge, it is necessary to employ advanced statistical techniques capable of detecting hidden variables and revealing complex dependencies within the data.

3. **Scalability:** Due to the rapid growth in data generation and collection, handling datasets that reach gigabyte, terabyte, or even petabyte scales has become a common necessity. To effectively manage these large volumes, data mining algorithms must be capable of scaling efficiently. Achieving scalability often involves designing specialized search strategies, creating advanced data structures for faster record retrieval, and implementing out-of-core algorithms for datasets that exceed memory capacity.Additionally, scalability can be further enhanced through sampling techniques and by developing parallel and distributed data processing methods.

4. **High dimensionality :** With the rapid advancement of technology, it has become common to encounter datasets containing hundreds or even thousands of attributes, compared to the limited number available decades ago. Fields such as bioinformatics, through microarray technologies, now produce gene expression data involving thousands of features. Similarly, datasets with temporal or spatial components, like repeated temperature measurements across locations, significantly increase the dimensionality. The presence of massive datasets and high-dimensional data creates combinatorially explosive search spaces, making it challenging for traditional data analysis techniques to perform effectively. Moreover, high dimensionality can lead to the identification of spurious patterns that lack general validity.To address these challenges, researchers employ solutions such as efficient algorithms, dimensionality reduction methods, sampling strategies, approximation techniques, massively parallel processing, and the integration of prior domain knowledge.

5. **Data security and privacy:**Data security and privacy are critical challenges in data mining, especially with the frequent handling of sensitive personal information. Organizations must comply with regulations like GDPR and HIPAA to avoid legal penalties and reputational damage. To mitigate risks of unauthorized access and misuse, strong security measures such as data encryption are essential. Encryption ensures that data

remains confidential and protected across databases, networks, and storage systems.

6. **Data quality:**Data quality is crucial in data mining, as poor data containing errors, inconsistencies, or missing values can lead to inaccurate and unreliable results. Maintaining data quality involves data cleansing to correct errors and remove duplicates, and data validation to ensure accuracy and reliability. High-quality data is essential for effective mining and trustworthy insights.[32][33][19][34][35][36]

1.6 Data warehousing

A Data Warehouse (DW) is a centralized system designed to collect, integrate, and store large volumes of structured and historical data from various sources. It serves as a unified repository that supports business intelligence (BI), advanced data analysis, and informed decision-making within organizations. Typically implemented using relational or multidimensional models—such as the star schema that organizes data into fact and dimension tables—data warehouses are often used alongside OLAP tools, enabling multidimensional analysis across axes like time, product, and location. In addition to their architectural role, data warehouses exhibit several key characteristics that distinguish them from traditional operational databases. They are subject-oriented, focusing on specific topics related to organizational domains. They are also integrated, combining data from heterogeneous sources into a consistent format. Once data is loaded, it becomes persistent and non-volatile, meaning it is not altered and remains stored for long-term analysis. Furthermore, data warehouses are time-variant and multi-temporal, containing both historical and current data to facilitate the analysis of trends and time series. They offer easy accessibility for end-users and are kept isolated from operational systems to ensure analytical stability and data integrity.

- **Subject-oriented data :** A data warehouse is organized around key subjects such as customers, sales, and products, with the aim of supporting analysis and decision-making rather than daily transactional operations. Data is collected from various systems regardless of their operational structure, providing a unified and simplified view of each subject while excluding irrelevant data that does not contribute to the decision support process.

- **Integrated data :** A data warehouse is built by integrating various data sources such as relational databases, flat files, and transaction records. This includes applying data cleaning and integration techniques to ensure consistency in naming conventions, encoding structures, and attribute measurements. Data is also standardized through normalization and the definition of a unified reference to ensure a consistent and comprehensive view of the information.

- **Time-variant data :** Data in a data warehouse is stored with a temporal dimension that enables tracking changes over extended periods. A unified time reference is adopted to ensure data is preserved and retrieved in its historical sequence, allowing for accurate analysis of information evolution over time.

- **Non-volatile data :** Data in a data warehouse is stable and not subject to direct modification. It is copied from operational systems and stored separately, eliminating the need for transaction processing or concurrency control. Access is typically limited to initial loading and querying, ensuring traceability of information and decisions over time.[37][38][39][40][41]

1.6.1 Data warehouse architecture components

A data warehouse consists of several interrelated core components designed to support analysis and decision-making processes within an organization. The main components include:

1. **Data sources:** These vary between internal and external data, such as operational databases, extended files (e.g., Excel), various enterprise systems, and market data. These data sources form the primary input for building the data warehouse

2. **ETL tools (Extract, Transform, Load)** These tools are central to the data preparation phase. First, data is **extracted** from various sources, then **transformed** by cleaning, aggregating, and adapting it for analytical purposes, and finally **loaded** into a target database. ETL tools vary in their techniques, flexibility in performing transformations, and their ability to handle incomplete data and apply data quality rules.

3. **Central database:** This serves as the foundational infrastructure of the data warehouse. It may consist of traditional databases or in-memory databases, which offer high speed and real-time responsiveness, especially in the context of big data challenges.

4. **Metadata:** Metadata is used to describe the data within the warehouse and is divided into two types: **technical metadata**, used by developers and system administrators to manage the warehouse, and **business metadata**, which adds context and meaning to the data and helps end users better understand it. Metadata plays a vital role in the development and maintenance of the data warehouse.

5. **Data marts:**These are smaller, specialized databases aimed at serving specific departments within the organization (e.g., sales or marketing). They derive their data from the central warehouse to facilitate focused analysis.

6. **Access and analysis tools:**These tools provide interactive interfaces for users to generate reports, analyze data from multiple perspectives using OLAP tools, perform data mining, and create interactive visualizations. These tools are often no-code, making them accessible to non-technical users.

7. **Data warehouse management:** This involves all administrative tasks related to operating, updating, and maintaining the data warehouse, including security, performance monitoring, and ensuring the integration and quality of stored data.

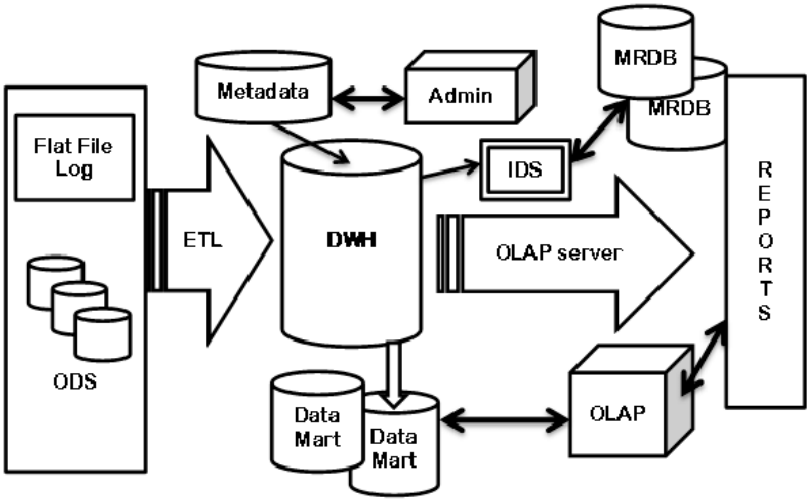


Figure 1.6: The structural design and main components of a data warehouse system

[40] [42] [43]

1.6.2 Comparative analysis: operational databases vs. data warehouses

OLTP (Online Transaction Processing) and OLAP (Online Analytical Processing) systems are considered fundamental in data management, albeit with distinct functionalities. OLTP focuses on the rapid processing of large volumes of transactions to ensure data consistency and reliability in daily business operations. Understanding data warehouses is facilitated by comparing them to the common OLTP systems. In contrast, OLAP is designed for complex data analysis and querying, enabling organizations to extract valuable insights from vast datasets to support strategic decision-making. [44] [45] The key distinguishing features between OLTP and OLAP are detailed below:

Feature	OLTP	OLAP
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB Design	ER-based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of Work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of Records Accessed	tens	millions
Number of Users	thousands	hundreds
Database size	GB to hundreds of GB	TB or more
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

Table 1.2: Comparison between OLTP and OLAP systems

1.6.3 Data warehouse architecture and its types

Data warehouse architecture is the strategic design framework for an organization's data storage system, aimed at efficiently analyzing diverse data. It involves the process of extracting and organizing data into a unified format to support complex analytical queries. It serves as a fundamental pillar of business intelligence, transforming raw data into insights that support strategic decision-making.

Types of data warehouse architectures

Data warehouse architecture is the structural design for organizing data in databases to extract valuable information from prepared raw data, often using a dimensional model to deliver business intelligence. It comprises three main types based on the number of tiers:

- **Single-tier data warehouse architecture** is a rarely implemented model. Its primary objective is to minimize redundancy by storing the least possible amount of data. How-

ever, its main drawback lies in the absence of a distinct component that separates analytical and transactional processing.

- **Two-tier data warehouse architecture** is characterized by the presence of a staging area that precedes the data warehouse layer. The primary function of this intermediate area, located between the data sources and the storage repository, is to ensure that all data undergoes the necessary cleansing and formatting processes before being loaded into the warehouse.

- **Three-tier data warehouse architecture** is the most common model, featuring three layers: a bottom database for cleansed data, a middle application layer with OLAP for analysis, and a top user interface for reporting and querying. Physically, it involves a source layer, a reconciled layer for integrated and cleansed data, and the data warehouse layer.

[42] [46]

1.6.4 A comparative analysis of data warehouses and data marts

A Data Warehouse (DW) and a Data Mart are both structured data repositories, but their scope differs. A DW is a central repository for the entire organization, while a Data Mart serves the needs of a specific department. A DW is designed as a structured relational database to optimize SQL queries. A Data Mart aims to isolate smaller data sets for easier access and can be created from a DW or other sources; multiple Data Marts can merge to form a DW. Key differences include that a Data Mart is department-specific, offers faster query speeds, supports tactical decisions, involves smaller and faster-changing data, and has a quicker implementation time compared to a DW, which supports strategic enterprise-wide decisions with broader data and longer implementation. [47] [48]

1.6.5 Multidimensional data model schemas: Star, snowflake, and fact constellation

The multidimensional data model is a core concept in data warehousing and OLAP, enabling data visualization and analysis across multiple dimensions (like product, time, and location). This model often materializes as a data cube, defined by dimensions and facts. Dimensions are the perspectives for recording data, typically linked to dimension tables providing detailed descriptions. Facts, on the other hand, are numerical measures used to analyze relationships between dimensions, stored in a central fact table. Dimensions inherently have a hierarchical nature, allowing for multi-level data analysis (e.g., years, quarters, months within a time dimension). In designing relational databases, the Entity-Relationship (ER) data model

is commonly used, where the database schema focuses on entities and their relationships, a suitable approach for Online Transaction Processing (OLTP). However, data warehouses necessitate a concise, subject-oriented schema that facilitates Online Analytical Processing (OLAP). The multidimensional data model is the most prevalent for data warehouses, typically manifesting in three primary schema types: star schema, snowflake schema, and fact constellation schema. Now, suppose we want to display the sales data with an additional third dimension. For example, we can view the data based on time and item, as well as location, for cities such as Chicago, New York, Toronto, and Vancouver. This 3-dimensional data is shown in Table 1.3, where it is represented as a series of two-dimensional tables. Conceptually, the same data can also be represented as a 3-D data cube, as illustrated in Figure 1.7. If we want to add a fourth dimension, such as supplier, the visualization becomes more complex. However, we can think of the 4-D cube as a sequence of 3-D cubes, as shown in Figure 1.8. If we continue...

Time	**Chicago**				**New York**				**Toronto**				**Vancouver**			
	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	728	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

Table 1.3: A 3-D view of sales data for allElectronics, according to the dimensions time, item, and location. The measure displayed is *dollars_sold* (in thousands).

Using this approach, we can represent any n-dimensional data as a collection of “cubes” with $(n - 1)$ dimensions. A data cube is a metaphor for storing multidimensional data, noting that the actual physical storage may differ from its logical representation. It is important to understand that data cubes are not limited to just three dimensions but can extend to n dimensions. Table 1.3 shows the data at different levels of summarization. In the field of data warehousing research, data cubes similar to those shown in Figures 1.7 and 1.8 are often referred to as "cuboids." Given a set of dimensions, a cuboid can be created for each subset of those dimensions, resulting in a hierarchical lattice of cuboids, where each cuboid represents a certain level of summarization or grouping. This hierarchical lattice is known as a data cube.

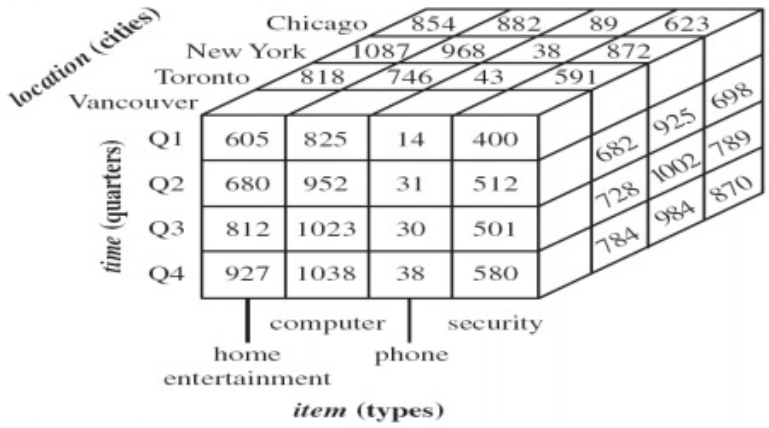


Figure 1.7: Multidimensional data cube representation: sales by location, item, and time

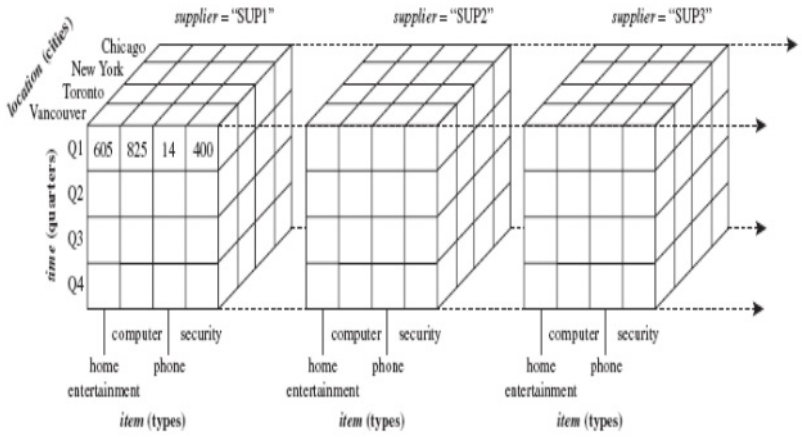


Figure 1.8: Four-dimensional data cube representation of allelectronics sales: analysis by time, item, location, and supplier (values displayed: dollars sold in thousands).

Star schema: The star schema is the most common model in designing multidimensional data warehouses. It is characterized by its centralized structure, consisting of:

- **A large, central fact table:** This table contains the numerical measures (facts) and represents the bulk of the data, designed to avoid redundancy.
- **A set of smaller dimension tables:** One table is dedicated to each analytical dimension, surrounding the central fact table and providing descriptive information.

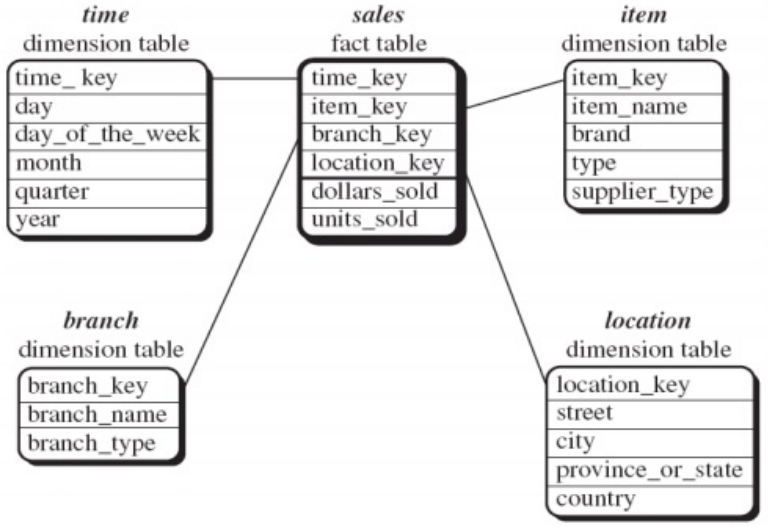


Figure 1.9: Star schema for a sales data warehouse

Snowflake schema The Snowflake Schema is considered an extension of the Star Schema, characterized by the normalization of certain dimension hierarchies. This process involves decomposing dimension tables into a set of smaller dimension tables, resulting in a graphical structure that resembles a snowflake shape. Although this design aims to reduce redundancy, its structure may reduce browsing effectiveness due to the need for more join operations when executing queries.

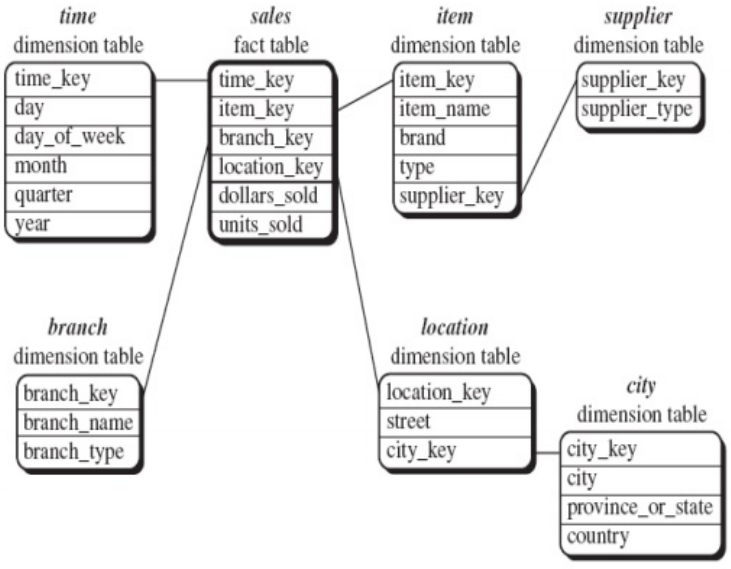


Figure 1.10: Snowflake schema for a sales data warehouse: illustration of fact and normalized dimension tables

Fact constellation schema Fact constellation schemas, also known as Galaxy Schemas, are essential for sophisticated applications that necessitate the sharing of dimension tables

among multiple fact tables. This type of schema can be viewed as a collection of interconnected star schemas, where fact tables share common dimension tables. This design enables comprehensive data analysis involving complex relationships and multiple measures.

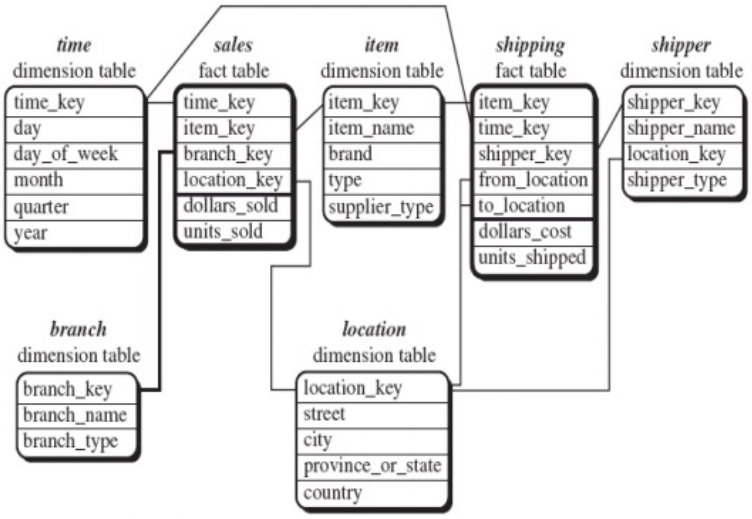


Figure 1.11: Fact constellation schema for an integrated sales and shipping data warehouse

Operations in multidimensional data model Operations in a multidimensional data model are fundamental for effective data analysis in data warehouses, allowing exploration from various levels and details.Key operations include:

- **Aggregation (roll-up):** Moving from a lower-level detail to a higher-level summary (e.g., total sales by city and year to total sales by region and year).
- **Selection (slice):** Defining a subcube by selecting a single value for a specific dimension (e.g., sales where city = Palo Alto and date = 1/15/96).
- **Drill-down:** The inverse of roll-up, moving from a high-level data summary to a lower level of detail, enabling deeper exploration of underlying patterns (e.g., sales minus expenses by city, or top 3% of cities by average income).
- **Visualization Operations:** Include, for example, Pivot or Dice, which rearrange and display data from different perspectives to enhance analysis.

These operations collectively enable users to effectively navigate data cubes and discover patterns and relationships within the data.

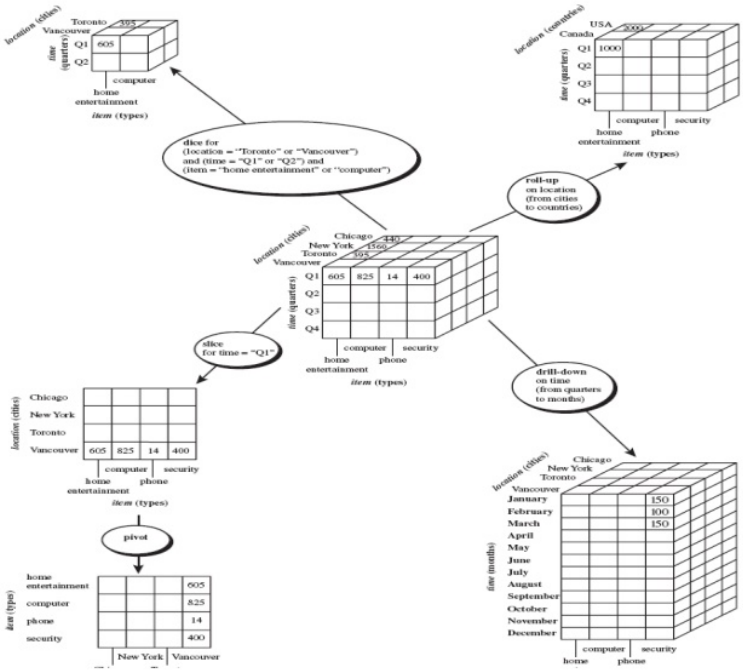


Figure 1.12: Online analytical processing (OLAP) operations on multidimensional data

[49] [50] [51]

1.6.6 Integration with data mining

Data integration in the context of data mining refers to the process of combining data from multiple and heterogeneous sources into a unified and consistent view. This is achieved through techniques such as data warehousing and ETL (Extract, Transform, Load) processes. The goal is to enhance data quality in terms of accuracy, consistency, and readiness for analysis. Integrating data mining systems with databases is a key step in improving data accessibility and enabling real-time analysis, which supports fast and informed decision-making. However, this integration presents technical challenges, such as ensuring data format compatibility, maintaining data integrity, managing system resources, and minimizing processing delays. When implemented effectively, this integration enhances operational efficiency, reduces data redundancy and errors, and enables the development of data-driven strategies that lead to more accurate and impactful business outcomes. [49] [19]

1.6.7 Role of data warehousing in kdd

Data warehousing serves as a foundational component in the Knowledge Discovery in Databases (KDD) process by offering a centralized and integrated repository of structured data. This infrastructure facilitates efficient data storage, retrieval, and preprocessing, which are es-

sential for enabling robust and scalable data mining operations. As such, data warehouses play a pivotal role in supporting the analytical phases of KDD and ensuring data quality and consistency across the process.

1.7 Conclusion

Throughout this chapter, it has become evident that Data Science cannot be separated from the broader ecosystem that includes Artificial Intelligence and Machine Learning, as they provide the intelligent tools necessary for sophisticated data analysis. Knowledge Discovery in Databases plays a critical role by offering a structured approach to transforming raw data into valuable insights, while Data Warehousing ensures the scalability and efficiency of data storage, enabling seamless access for both real-time and historical analysis. Together, these concepts form an integrated foundation for building intelligent information systems capable of supporting decision-making and predictive analytics. Their importance goes beyond technical implementation; they are now strategic assets that empower organizations to understand customer behavior, optimize operations, and adapt swiftly to dynamic environments. As such, mastering these domains is not merely an option, but a necessity for anyone seeking to innovate or operate effectively in a data-centric world.

Chapter 2: Clustering

2.1 Introduction

As organizations continue to accumulate massive volumes of data, the challenge lies not only in storing this data but in uncovering hidden patterns and meaningful insights within it. Data Mining emerges as a crucial process in this context, offering techniques and methodologies to explore large datasets and extract valuable knowledge that can inform decision-making. Often described as the core step within the broader framework of Knowledge Discovery in Databases (KDD), data mining focuses on discovering patterns, correlations, anomalies, and trends that are not immediately apparent through traditional analysis.
This chapter explores the foundational concepts of data mining, beginning with an overview of the data mining process, which includes stages such as data selection, preprocessing, transformation, and pattern evaluation. Additionally, it delves into various data mining techniques, with particular emphasis on clustering, an unsupervised learning approach that groups similar data points based on specific criteria. Finally, the chapter discusses the types of cluster analysis, presenting their characteristics, applications, and the challenges associated with implementing them in real-world scenarios.

2.2 Data mining

Data mining is an automated analytical process aimed at extracting valuable, previously unknown information from large amounts of data. This is achieved using mathematical, statistical techniques, and artificial intelligence algorithms, with the goal of uncovering hidden patterns and relationships that contribute to scientifically and efficiently supporting decision-making. [52] [53]

2.2.1 the process of data mining

The data mining process typically involves the following steps:

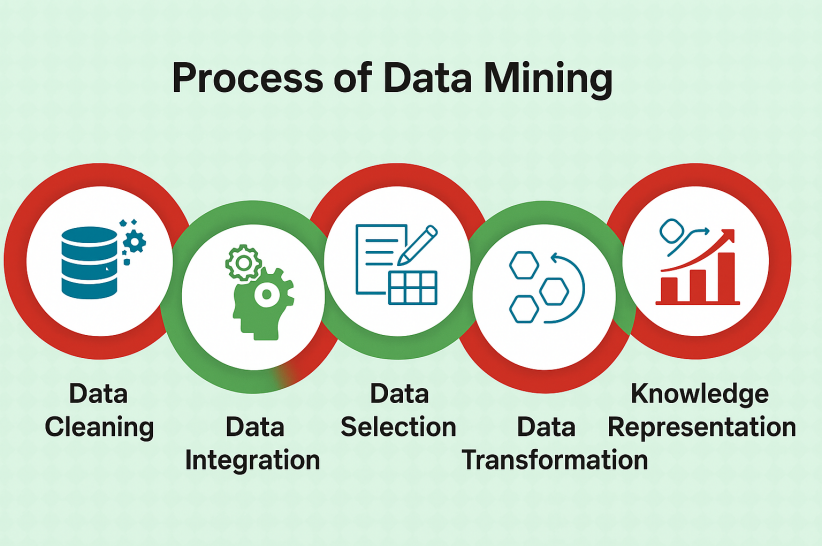


Figure 2.1: the process of data mining

[54]

Data cleaning (or data cleansing)

The first step in the KDD process, data cleaning, involves the identification and removal of noise, inconsistencies, and irrelevant data from the dataset. This phase ensures that the data used for analysis is of high quality and free from errors, which can lead to inaccurate findings.

Data integration

In this phase, multiple heterogeneous data sources are integrated into a unified dataset. This step may involve combining data from different databases, data warehouses, or external datasets. The goal is to create a consolidated data source that can be analyzed in a consistent manner.

Data selection

Data selection refers to the process of identifying and retrieving relevant data from the larger dataset. In this phase, the focus is on selecting only the data that is pertinent to the problem at hand, which helps streamline the analysis and reduce the complexity of the data mining process.

Data transformation

Also referred to as data consolidation, this phase involves transforming the selected data into a format that is suitable for mining. It may include normalizing values, aggregating

data, encoding categorical variables, or other necessary data manipulations to ensure that the dataset is in an appropriate form for the modeling phase.

Knowledge representation

The final phase, knowledge representation, involves presenting the discovered knowledge in a format that is understandable and actionable. Visualization techniques, such as charts, graphs, or dashboards, are often employed to help users interpret and make sense of the findings. This step is essential for facilitating the application of the extracted knowledge to real-world problems.

2.2.2 Applications of data mining

Data mining techniques have proven to be highly effective across various practical domains, leading to measurable and impactful benefits. The following overview highlights some of the key application areas where data mining plays a vital role in extracting valuable insights from large datasets. [55]

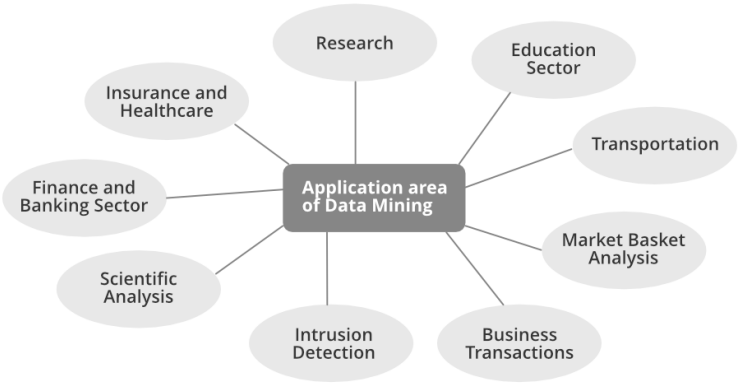


Figure 2.2: Applications of data mining

Scientific analysis

Scientific fields like nuclear physics and psychology generate huge amounts of data through simulations and experiments. Data mining techniques help analyze this complex data efficiently, allowing scientists to discover new patterns and insights faster than traditional analysis methods.

Intrusion detection

Data mining is widely used in cybersecurity to detect network intrusions. It helps classify and extract useful information from large datasets to identify abnormal behaviors, potential

threats, and unauthorized activities. These techniques support Intrusion Detection Systems (IDS) by improving the accuracy of alerts and recognizing patterns of attacks.

Business transactions

All business transactions, whether internal or external, are time-sensitive and data-rich. Data mining helps businesses analyze these transactions to discover trends, consumer behaviors, and to support quick, competitive decision-making in a constantly changing market.

Market basket analysis

This technique identifies purchasing patterns in retail environments. By analyzing which products are frequently bought together, businesses can optimize product placement, promotions, and cross-selling strategies. Data mining enables deeper insight into consumer habits.

Education (educational data mining – edm)

In education, data mining helps analyze student behaviors, performance, and learning patterns. These insights support educators in customizing teaching methods and improving academic outcomes for students.

Research

In scientific and technical research, data mining is used for prediction, classification, clustering, and finding associations. By creating and testing models, researchers can extract valuable knowledge from large datasets, enhancing the precision and quality of their studies.

Healthcare and insurance

In healthcare, pharmaceutical companies use data mining to improve targeting of doctors and predict the effectiveness of marketing campaigns. In insurance, it helps identify customer segments, predict policy purchases, and detect fraudulent claims based on behavioral patterns.

Transportation

Data mining helps transportation companies improve logistics, identify promising customers, and optimize sales strategies based on historical service usage and customer profiles.

Financial and banking sector

Banks and financial institutions apply data mining to customer transaction records to identify individuals likely to adopt new credit products or services. It enhances marketing campaigns and reduces financial risks.

2.2.3 Data mining techniques

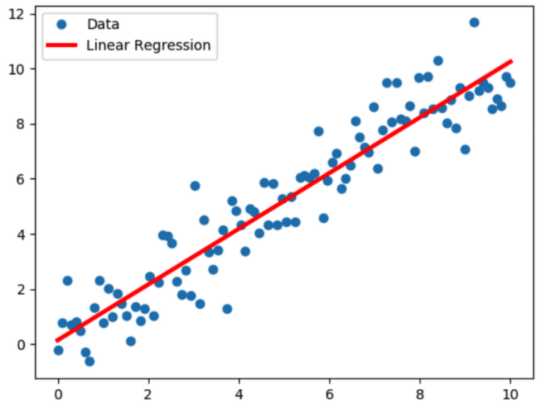
There are a wide array of data mining techniques used in data science and data analytics. Your choice of technique depends on the nature of your problem, the available data, and the desired outcomes. Predictive modeling is a fundamental component of mining data and is widely used to make predictions or forecasts based on historical data patterns. You may also employ a combination of techniques to gain comprehensive insights from the data. Top-10 data mining techniques: [56]

Classification

Classification is a technique used to categorize data into predefined classes or categories based on the features or attributes of the data instances. It involves training a model on labeled data and using it to predict the class labels of new, unseen data instances.

Regression

Regression is employed to predict numeric or continuous values based on the relationship between input variables and a target variable. It aims to find a mathematical function or model that best fits the data to make accurate predictions.



Source: ResearchGate

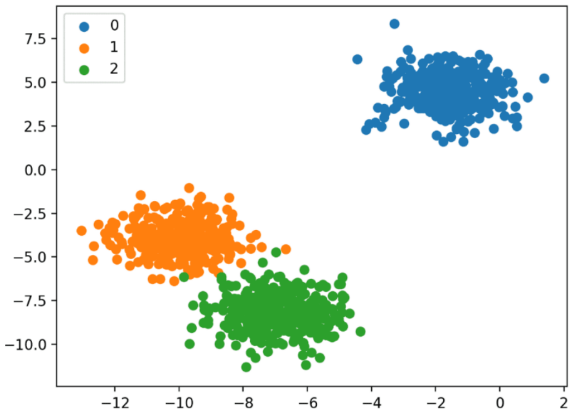
Figure 2.3: Regression

- **??** illustrates a Simple Linear Regression model, a fundamental technique in supervised learning used to predict the value of a dependent variable based on an independent

variable. The blue dots represent the actual data points, while the red line indicates the best-fit regression line learned from the data. The goal of linear regression is to find the optimal linear relationship between the two variables, enabling the model to make accurate predictions on new, unseen data. It is widely applied in fields like economics, healthcare, and data analysis for forecasting and decision-making.

Clustering

Clustering is a technique used to group similar data instances together based on their intrinsic characteristics or similarities. It aims to discover natural patterns or structures in the data without any predefined classes or labels.



Source: Machine Learning Mastery

Figure 2.4: Clustering

- Figure 2.4 shows the result of applying the K-Means clustering algorithm to a two-dimensional dataset. It is an example of unsupervised learning, where the algorithm groups the data into three distinct clusters without prior knowledge of the labels. The colors (blue, orange, and green) represent the cluster each data point belongs to, demonstrating the algorithm's ability to identify patterns and structure within the data.

Association rule

Association rule mining focuses on discovering interesting relationships or patterns among a set of items in transactional or market basket data. It helps identify frequently co-occurring items and generates rules such as "if X, then Y" to reveal associations between items. This simple Venn diagram shows the associations between itemsets X and Y of a dataset.

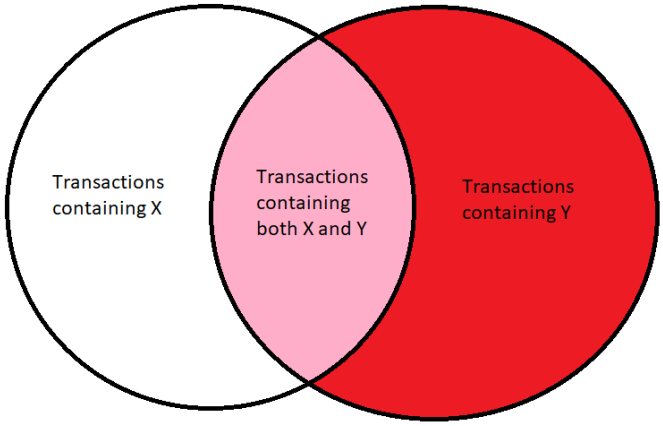


Figure 2.5: Association Rule

- Figure 2.5represents a Venn Diagram used in association rule analysis, illustrating the relationship between transactions that contain item X and those that contain item Y. The left circle represents all transactions that include X, while the right circle represents those that include Y. The overlapping area between the two circles shows the transactions that contain both X and Y. This visual representation helps in understanding the strength of association between items and is commonly used to calculate key metrics such as support and confidence in data mining.

Anomaly detection

Anomaly detection, sometimes called outlier analysis, aims to identify rare or unusual data instances that deviate significantly from the expected patterns. It is useful in detecting fraudulent transactions, network intrusions, manufacturing defects, or any other abnormal behavior.

Time series analysis

Time series analysis focuses on analyzing and predicting data points collected over time. It involves techniques such as forecasting, trend analysis, seasonality detection, and anomaly detection in time-dependent datasets.

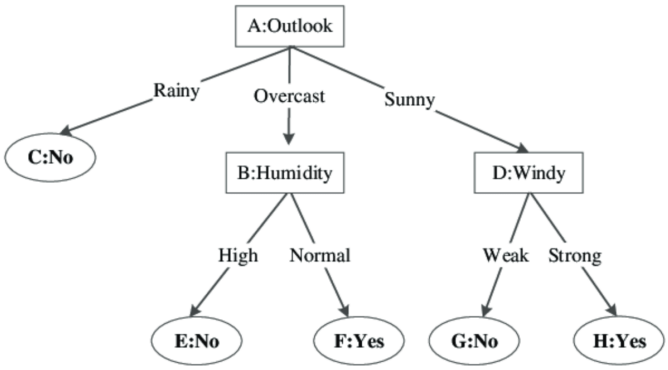
Neural networks

Neural networks are a type of machine learning or AI model inspired by the human brain's structure and function. They are composed of interconnected nodes (neurons) and layers that can learn from data to recognize patterns, perform classification, regression, or other tasks.

Decision trees

Decision trees are graphical models that use a tree-like structure to represent decisions and their possible consequences. They recursively split the data based on different attribute values to form a hierarchical decision-making process.

The image shows a decision tree used for classification. It starts with A: Outlook, which checks if the weather is Rainy, Overcast, or Sunny. If it is Rainy, the decision is No (C). If it is Overcast, we check Humidity (B): if it is High, the decision is No (E), and if it is Normal, the decision is Yes (F). If it is Sunny, we move to Windy (D): if the wind is Weak, the decision is No (G), but if it is Strong, the decision is Yes (H). Each complete path from the root to the final decision shows how the attributes Outlook, Humidity, and Windy help in making the classification.



Source: ResearchGate

Figure 2.6: Decision Trees

- Figure 2.6 represents a Decision Tree, a widely used model in machine learning for classification and decision-making tasks. The tree starts at the root node labeled Outlook, which splits the data based on weather conditions (Rainy, Overcast, Sunny). Each branch leads to either a decision (Yes or No) or another feature node such as Humidity or Windy, which further splits the data. For instance, if the outlook is overcast, the decision depends on humidity levels, while if it is sunny, the decision is based on wind strength. The leaf nodes (C, E, F, G, H) represent the final classification outcomes. This structured approach allows the model to make interpretable and rule-based predictions, often used in scenarios requiring clear reasoning and transparency.

Ensemble methods

Ensemble methods combine multiple models to improve prediction accuracy and generalization. Techniques like Random Forests and Gradient Boosting utilize a combination of weak learners to create a stronger, more accurate model.

Text mining

Text mining techniques are applied to extract valuable insights and knowledge from unstructured text data. Text mining includes tasks such as text categorization, sentiment analysis, topic modeling, and information extraction, enabling your organization to derive meaningful insights from large volumes of textual data, such as customer reviews, social media posts, emails and articles.

2.3 Clustering

Clustering, or cluster analysis, is the process of automatically partitioning a set of data objects into clusters, where objects within the same cluster are similar to each other and different from those in other clusters. It helps uncover hidden patterns, dense and sparse regions, and relationships between data attributes. Clustering is widely used in applications such as market research, biology, image processing, fraud detection, and document classification. It can serve as both a standalone tool for exploring data and as a preprocessing step for further data mining tasks. [49] [57] [58]

2.3.1 Types of cluster analysis

Cluster analysis is the process of grouping objects based on their similarities. There are various clustering algorithms, and the choice of method depends on factors like dataset size, dimensionality, and the number of clusters. It's important to note that an algorithm effective for one dataset may not work for another. Therefore, selecting the right method often requires experimentation. [59] [60]

Hierarchical clustering

Hierarchical Clustering is a method used to group data based on the proximity and connectivity of their attributes. It can be performed using two main strategies: **Agglomerative** and **Divisive**.

1. **Agglomerative hierarchical clustering** This method follows a **bottom-up** strategy. It begins by assigning each object to its own cluster and iteratively merges the closest clusters into larger clusters, until all objects are in a single cluster or certain termination conditions are met. The merging step identifies the two closest clusters (based on a similarity measure) and combines them into one cluster. Because two clusters are merged in each iteration, the algorithm requires at most $n - 1$ iterations. The user can specify the desired number of clusters as a termination condition. One challenge with

agglomerative methods is **chaining** , where larger clusters tend to attract more points, causing the clusters to grow unbalanced.

2. **Divisive hierarchical clustering :** In contrast,**divisive clustering** employs a **top-down** strategy. It starts by placing all objects in a single cluster, which is the root of the hierarchy. Then, this root cluster is recursively partitioned into smaller sub-clusters. The partitioning process continues until the clusters at the lowest level are coherent, either containing only one object or containing objects that are sufficiently similar. This approach is more flexible in terms of both the hierarchical structure and the balance of the clusters. It can be faster than agglomerative methods, particularly when the tree structure does not need to be constructed all the way down to individual data points.

Partitioning clustering algorithms

Partitioning clustering divides a set of N objects into k clusters, optimizing a criterion function. Each cluster is represented by either its **centroid** (e.g., **k-means**) or its **medoid**(e.g., **k-medoids**). The algorithm typically starts with k randomly selected seeds, then iteratively reassigns points to minimize the clustering criterion, commonly the **square-error criterion**, which reduces the sum of squared Euclidean distances between points and their closest centroid. The main goal is to divide the data into k partitions, each representing a cluster. However, a major drawback is that partitioning methods can produce different solutions based on initial conditions. Additionally, these methods may perform poorly when clusters overlap or when points from different clusters are very close together. The total number of possible partitions $P(n, k)$ for dividing n patterns into k clusters is:

$$P(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-1}\frac{k}{i}i^n \tag{2.1}$$

[61] [62] **Types of partitioning algorithms :** There are four main types of partitioning algorithms: the K-Means algorithm, the K-Medoid algorithm (also known as PAM - Partitioning Around Medoids), and the CLARA and CLARANS algorithms.

1. **K-means algorithm:** The K-Means algorithm was first developed by James Macqueen in 1967. In this method, a cluster is represented by its centroid, typically the mean of the points within the cluster. The objective function used in K-Means is the sum of discrepancies between a point and its centroid, usually expressed through an appropriate distance measure. The clusters formed are convex in shape.

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|X_i^j - C_j\right\|^2 \tag{2.2}$$

Procedure of K-Means

(a) Arbitrarily select k objects from the data set D (which contains n objects) as the initial cluster centers, where k is the number of clusters.

(b) Repeat the first step.

(c) Reassign each object to the cluster whose centroid it is most similar to, based on the mean value of the objects in the cluster.

(d) Calculate the new mean for each cluster.

(e) Repeat the steps until no further changes occur.

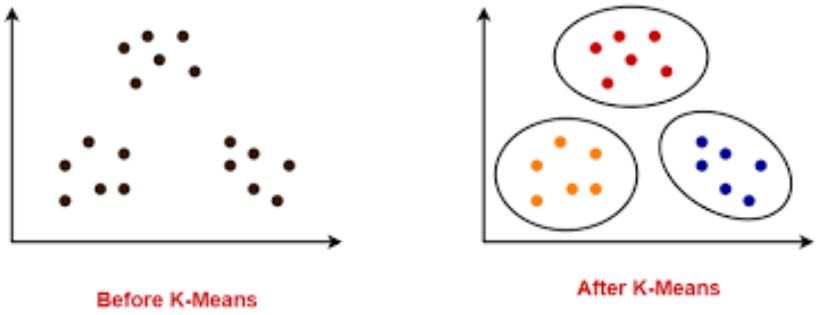


Figure 2.7: Visualization of k-means clustering process

This figure demonstrates the effect of the K-Means clustering algorithm. The left side shows unstructured data before clustering, while the right side illustrates how K-Means partitions the data into distinct groups based on similarity. Each cluster is represented with a different color, highlighting the algorithm's ability to organize data into well-separated and cohesive groups. [61] [62]

- **K-medoids algorithm** The K-Medoids algorithm is a method for partitioning data into k clusters. Each cluster is represented by a central object called a "medoid." This algorithm is more resistant to outliers than K-Means because it uses actual data points as cluster centers instead of means. PAM (Partitioning Around Medoids) is an example of a K-Medoids algorithm, which starts by selecting random medoids and then iteratively tries to improve them to minimize the dissimilarity between the objects and their medoids. Despite its effectiveness, K-Medoids can be computationally expensive.

[61] [62] [63] **K-medoids algorithm: simplified procedure**

1. **Start:** Randomly select K objects as initial medoids.
2. **Assign:** Associate each object to the most similar medoid.
3. **Improve:**

- Randomly select a non-medoid object.
- Calculate the cost of swapping it with a current medoid.
- If the cost improves, perform the swap.

4. **Repeat:**Continue the improvement process until the medoids no longer change.
    - **Goal:**To form K clusters represented by medoids that minimize the dissimilarity within the clusters.

- **CLARA (clustering large applications)** CLARA is a clustering algorithm that extends the K-Medoids algorithm, specifically designed for handling large datasets. While K-Medoids requires calculating pairwise distances between all data points, making it computationally expensive for large data, CLARA addresses this by selecting a small sample of the data to choose medoids, significantly reducing computation time. Developed by Kaufman and Rousseeuw in 1990, CLARA applies the PAM (Partitioning Around Medoids) algorithm to multiple samples of the data (typically of size $40 + 2k$). It then assigns each object in the entire dataset to the closest medoid found in one of the samples. The average dissimilarity for each resulting clustering is calculated, and the set of medoids that yields the lowest average dissimilarity is retained as the best set of medoids. This process is repeated several times to improve the clustering quality.

[61]

- **CLARANS: large data clustering via randomized search** CLARANS is a method for clustering large datasets that combines the sampling approach of CLARA and the logic of PAM. The clustering process is viewed as a search for the best set of k medoids, where possible solutions are explored by replacing current medoids. The algorithm searches for better solutions (lower squared error) by comparing the current solution to a set of neighboring solutions. If a better solution is found, it moves to it and repeats the process. Upon reaching a locally good solution, it restarts the search with another random starting point. This randomized search increases the chances of finding high-quality clusterings, but the algorithm's performance depends on some user-defined settings.

[64] [62]

Density-based clustering

Aim to discover clusters with arbitrary shapes by identifying dense regions of objects and separating them from low-density regions (noise). They rely on grouping neighboring objects if their local density exceeds a certain threshold. These methods can discover non-regularly

shaped clusters, handle outliers well, and are scalable. There are two main approaches: one links density to a data point (e.g., DBSCAN and OPTICS), and the other links density to a point in the attribute space (e.g., DENCLUE). Challenges include handling regions with significantly different densities within the same cluster and occasional difficulty in interpreting results, as well as their performance in high-dimensional spaces.

1. **DBSCAN (density-based spatial clustering of applications with noise)** A density-based clustering algorithm for discovering arbitrarily shaped clusters in spatial data with noise. It identifies clusters as high-density regions of connected points. It relies on defining core points that have a sufficient number of neighbors within a certain distance. Non-core points within the neighborhood of core points are the cluster boundaries, and the rest are considered noise. It is characterized by its ability to handle non-regular shapes and outliers, but it is sensitive to distance and neighbor count parameters and may face difficulties in high-dimensional data. The computational complexity is $O(n^2)$ or $O(n \log n)$ with spatial indexing. **DBSCAN algorithm procedure**

   (a) Arbitrarily select a point r.

   (b) Retrieve all points density-reachable from $r$with respect to ϵ (Eps) and MinPts.

   (c) If r is a core point, a cluster is formed.

   (d) If r is a border point, no points are density-reachable from r, and DBSCAN visits the next point of the database.

   (e) Continue the process until all of the points have been processed

   Algorithm Mechanis

   (a) Classify each point as core, border, or noise.

   (b) Link nearby core points (within ϵ)

   (c) Form clusters from connected core points.

   (d) Assign border points to nearby clusters.

   Result: Clustering of density-connected points and noise identification

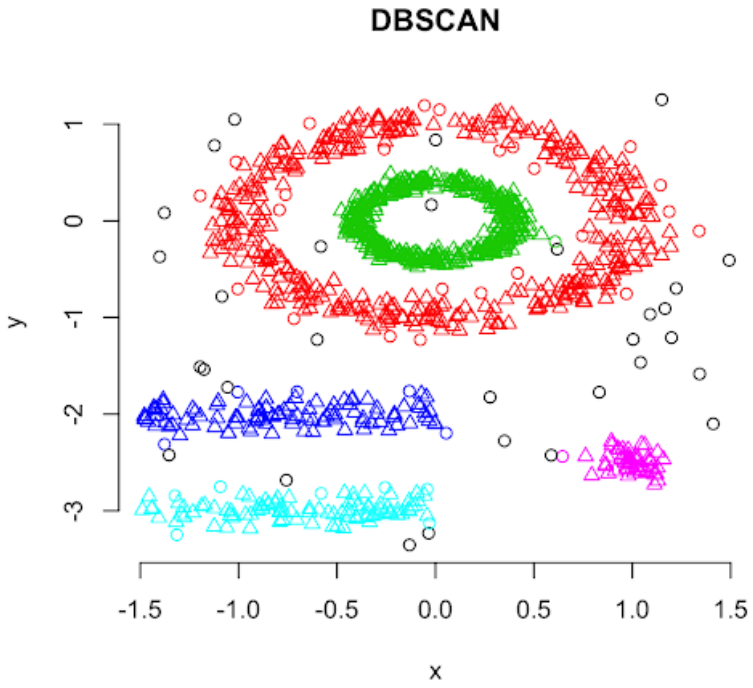


Figure 2.8: DBSCAN Clustering Result with Noise.

This figure illustrates the output of the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm. Unlike traditional clustering techniques such as k-means, DBSCAN is capable of discovering clusters of arbitrary shapes and identifying noise or outlier points that do not belong to any cluster. In this visualization, different symbols and colors represent distinct clusters, while scattered black circles indicate noise. The algorithm effectively separates dense regions from sparse ones without requiring prior knowledge of the number of clusters, making it particularly useful in applications involving spatial or irregular data distributions. [64] [61] [49] [65] [46]

2. **The OPTICS (Ordering points to identify the clustering structure)** algorithm extends the concept of density-based clustering. It computes an augmented cluster ordering for automatic and interactive cluster analysis. Due to the structural equivalence of the OPTICS algorithm to DBSCAN, OPTICS shares the same runtime complexity. Consequently, the runtime complexity of the OPTICS algorithm is $O(n \log n)$ if a spatial index is utilized, where n represents the number of objects in the dataset. This augmented ordering enables a more detailed exploration of the inherent clustering structure within the data. [63]

3. **DENCLUE (Density-based clustering)** DENCLUE is a clustering method that relies on defining density in the data space using influence functions for each point. The overall density is calculated as the sum of these functions, and clusters are determined

by finding the local maxima of density ("density attractors"). The algorithm uses a grid structure to efficiently compute density and is characterized by its scalability, ability to discover arbitrary shapes, noise resistance, and insensitivity to data order. However, it is sensitive to input parameters and negatively affected by high dimensionality, and its results may be difficult to interpret. **The algorithm operates through the following key steps**

(a) **Density function calculation:** The influence of each data point contributes to the overall density function of the data space.

(b) **Density attractor identification:** A hill-climbing procedure is employed to find these local density maxima. Starting from a point, the algorithm iteratively moves to neighbors with higher density until a local peak is reached.

(c) **Cluster formation:** Points are assigned to the cluster of the density attractor they are drawn to during the hill-climbing process or if they are within a defined neighborhood of the attractor.

(d) **Noise reduction (optional):** Clusters associated with density attractors below a certain minimum density threshold (*mindens*) can be considered noise and discarded.

(e) **Cluster merging (optional):**Clusters whose density attractors are connected by a path of points with density above (*mindens*) may be merged.

[63] [61]

Model-based clustering methods

Rely on assuming a mathematical model for each cluster to find the best representation of the data. They can automatically determine the number of clusters based on statistics. They may use a density function to locate clusters and provide a way to determine their number while considering outliers, resulting in robust clustering. Statistical approaches and COBWEB are examples. [62]

Grid-based clustering methods

These methods divide the object space into defined grid cells and perform clustering operations on these cells. Their main advantage is high processing speed that depends on the number of cells, not the number of objects. They are used efficiently in spatial data mining and can be integrated with other methods like density-based and hierarchical clustering. Examples include STING and CLIQUE. [49] [66]

- **STING (Statistical information grid)** is a grid-based multiresolution clustering technique where the embedding spatial area of the input objects is divided into rectangular cells. This space can be partitioned hierarchically and recursively, resulting in different levels of resolution that form a hierarchical structure. Statistical information regarding the attributes within each grid cell, such as the mean, maximum, and minimum values, is precomputed and stored as statistical parameters useful for query processing and other data analysis tasks.
- **CLIQUE (Clustering in quest)** is a simple grid-based method for finding density-based clusters in subspaces. CLIQUE partitions each dimension into non-overlapping intervals, thereby dividing the entire embedding space of the data objects into cells. It uses a density threshold to identify dense and sparse cells. A cell is considered dense if the number of objects mapped to it exceeds the density threshold. The main strategy behind CLIQUE for identifying a candidate search space relies on the monotonicity property of dense cells with respect to dimensionality, which is based on the Apriori property used in frequent pattern and association rule mining. **Grid-based clustering methods typically involve the following systematic steps:**
  - **Discretize:** Divide the data space into a finite grid of cells.
  - **Iterate:** Process each unvisited cell.
  - **Assess density:** Calculate the cell's density.
  - **Cluster formation:** If density exceeds a threshold, mark as a new cluster and expand to dense neighbors iteratively.
  - **Repeat:** Continue until all cells are processed.
  - **Terminate:** end the algorithm.

2.3.2 Condensed requirements for effective cluster analysis

Effective cluster analysis necessitates algorithms that address key challenges such as scalability, data diversity, discovery of various shapes, reduced reliance on prior parameters, noise resistance, support for incremental updates and insensitivity to input order, handling high dimensionality, constraint integration, and most importantly, producing interpretable and usable results. Meeting these requirements is essential for developing robust and practical clustering tools. [49]

2.3.3 Comparative analysis of clustering algorithms in data mining

Comparative study of clustering algorithms, focusing on density-based methods for selection Table 2.1 [63] [67]

Table 2.1: Comparison of Clustering Methods

Clustering Method	Advantages	Disadvantages
Partitioning	• Relatively scalable and simple. • Suitable for datasets with compact spherical clusters that are well-separated.	• Degradation in high dimensional spaces. • Poor cluster descriptors. • High sensitivity to initialization phase, noise and outliers.
Hierarchical	• Embedded flexibility regarding the level of granularity. • Well suited for problems involving point linkages, e.g. taxonomy trees. • Application to any attribute types.	• Inability to make corrections once the splitting/merging decision is made. • Lack of interpretability regarding the cluster descriptors. • Vagueness of termination criterion. • Prohibitively expensive for high dimensional and massive datasets.
Density based	• Discovery of arbitrary-shaped clusters with varying size. • Resistance to noise and outliers.	• High sensitivity to the setting of input parameters. • Poor cluster descriptors. • Unsuitable for high-dimensional datasets.

2.3.4 Applications of clustering

Clustering is a fundamental technique in data mining and machine learning, employed to uncover hidden structures within unlabeled data across diverse fields. Its prominent applications include:

1. Pattern recognition: Identifying groups of similar objects to discover underlying patterns and relationships.

2. Spatial data analysis: Grouping adjacent points or regions with similar characteristics for planning and geographical studies.

3. Image processing: Utilized in image compression (via color quantization) and object segmentation.

4. Bioinformatics: Grouping similar genes or proteins to reveal biological relationships.

5. Economic sciences and market research: Notably, customer segmentation for categorizing clients based on behavior and preferences, enabling tailored marketing strategies.

6. Document classification and text analysis: Sorting and grouping similar texts to enhance organization and search efficiency.

7. Web log data analysis: Discovering groups of similar access patterns to understand user behavior.

These applications highlight the practical value of clustering in extracting knowledge from data. [61] [68]

2.3.5 Types of linkages in hierarchical clustering

Hierarchical clustering organizes data into a tree-like structure by grouping similar points. Key to this process is Linkage, which calculates the distance between clusters before they are merged or divided, influencing the resulting cluster shapes and the dendrogram. Common linkage types include:

- **Single linkage** Single linkage, also known as minimum linkage, in hierarchical clustering is defined as the measure of distance between two clusters R and S based on the minimum distance between any two points, one from cluster R and the other from cluster S. Mathematically, the distance between the two clusters d(R,S) can be expressed as follows:

$$L(R,S) = \min\left(D(i,j)\right), \quad i \in R,\ j \in S \tag{2.3}$$

  where

- D(i, j): Distance function between points i and j.

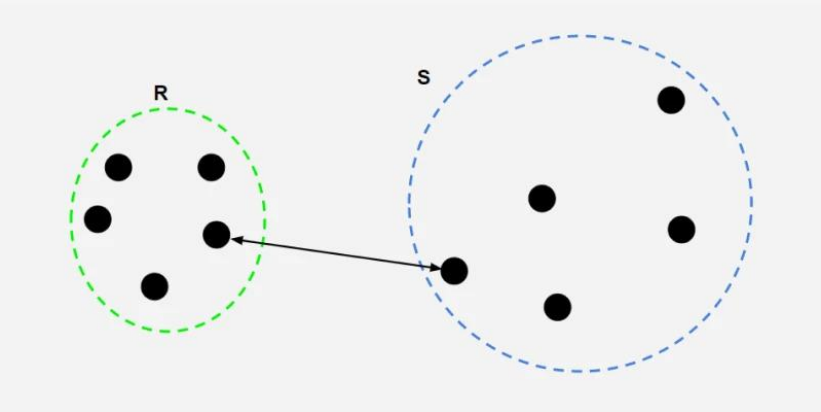


Figure 2.9: The principle of single-linkage in hierarchical clustering

This figure demonstrates the single linkage method, where the distance between two clusters is the minimum distance between any single point in one cluster and any point in the other. It is efficient but may result in elongated or "chained" clusters due to its sensitivity to close individual points.

- **Complete linkage** Complete linkage, also known as maximum linkage, in hierarchical clustering is defined as the measure of distance between two clusters R and S based on the maximum distance between any point from cluster R and any point from cluster S. This method tends to produce compact and spherical clusters due to its sensitivity to outliers, ensuring that all points within a single cluster are close to each other, thus forming cohesive clusters.Mathematically, the distance between the two clusters d(R,S) can be expressed as follows:

$$L(R,S) = \max\left(D(i,j)\right), \quad i \in R,\ j \in S \tag{2.4}$$

where

- D(i, j): Distance function between points i and j.

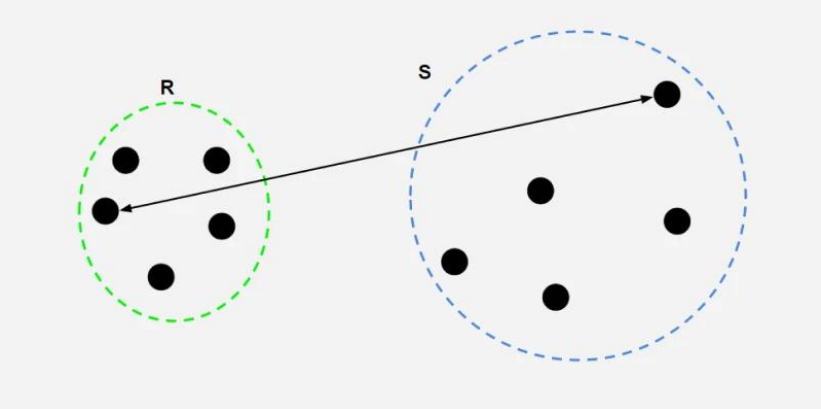


Figure 2.10: Representation of complete linkage between clusters R and S

This figure illustrates the complete linkage method in hierarchical clustering. In this approach, the distance between two clusters is defined as the maximum distance between any pair of points—one from each cluster. It tends to create compact and spherical clusters but is sensitive to outliers.

- **Average linkage** Average linkage in hierarchical clustering is defined as the measure of distance between two clusters R and S based on the average of the distances between all possible pairs of points, where each point of a pair is taken from a different cluster. This method balances the characteristics of single linkage and complete linkage by considering all pairs of points, not just the closest or farthest. Average linkage tends to produce more spherical and balanced clusters compared to single linkage, with less sensitivity to outliers compared to complete linkage, making it a popular choice in many applications. Mathematically, the distance between the two clusters d(R,S) can be expressed as follows:

$$L(R,S) = \frac{1}{n_R \times n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), \quad i \in R,\ j \in S \tag{2.5}$$

where

- n_R: Number of data-points in R
- n_S: Number of data-points in S

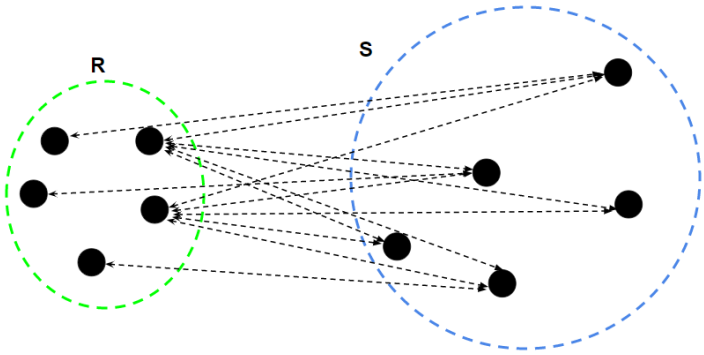


Figure 2.11: Representation of average linkage between clusters R and S

This figure shows the average linkage method, which calculates the average distance between all pairs of points from the two clusters. It provides a balance between single and complete linkage methods, offering more stable clustering results.[69][70]

2.4 Conclusion

In conclusion, data mining represents a powerful toolset for extracting actionable insights from complex and voluminous data sources. Through a structured process, it enables the identification of patterns and relationships that support intelligent decision-making across multiple domains. Among the techniques discussed, clustering plays a pivotal role by enabling the discovery of natural groupings within data without prior labeling, making it especially useful in exploratory data analysis and segmentation tasks. The chapter has shown that understanding the principles and methodologies of data mining, as well as mastering clustering techniques and their variations, is essential for effectively navigating today's data-driven environments. As organizations increasingly rely on data to guide their strategies, the ability to implement and interpret data mining outcomes will remain a critical competency for professionals and researchers alike.

Chapter 3: Comparison and results

3.1 Introduction

This chapter presents an analytical and comparative study of several clustering algorithms, with a particular focus on density-based methods such as DBSCAN, DPC, and HDBSCAN, alongside benchmark algorithms like K-means and GBC. The main objective of this study is to assess the effectiveness of each algorithm in classifying unlabeled data by applying them to both real-world and synthetically generated datasets. To accurately evaluate their performance, a range of well-established quantitative metrics were used, including NMI, V-measure, Silhouette Score, Dunn Index, and Davies-Bouldin Index, in addition to execution time as a practical performance indicator. The results of these experiments were analyzed and presented through comparison tables and illustrative bar charts, enabling an objective and detailed comparison of the algorithms in terms of quality and efficiency.

3.2 Clustering algorithm evaluation metrics

Clustering algorithm evaluation metrics are used to measure the quality and efficiency of clustering results. External metrics such as Normalized Mutual Information (NMI) and V-measure assess the similarity between the clustering results and the ground truth. Execution time is also considered an important metric to evaluate the efficiency of an algorithm. On the other hand, internal metrics like the Silhouette Score, Dunn Index, and Davies-Bouldin Index evaluate the clustering quality based on the internal structure of the data without requiring ground truth labels. [49] [71]

3.2.1 Normalized mutual information (NMI)

Normalized Mutual Information (NMI) is an external metric for evaluating the performance of clustering algorithms. It normalizes the Mutual Information (MI) score to scale results between 0 (no mutual information) and 1 (perfect correlation). NMI is calculated by normalizing MI using a generalized mean of the true and predicted label entropies. As an external metric, NMI requires the availability of ground truth class labels to assess the similarity between two partitions, with higher values indicating greater similarity.The NMI is calculated according to the following equation:

$$NMI(X,Y) = \frac{2 \cdot MI(X,Y)}{H(X) + H(Y)} \tag{3.1}$$

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{3.2}$$

where $H(X)$ and $H(Y)$ denote the Shannon entropy of partitions X and Y, and $H(X,Y)$ represents their joint entropy. [72] [73]

3.2.2 V-measure

The v-measure, as referenced in the Scikit-learn documentation (scikit-learn, n.d.), is an external evaluation metric for clustering algorithms. This metric aims to provide a balanced assessment of how well the discovered clusters align with the true class labels (ground truth) of the data.
Mathematically, the V-measure is defined as the harmonic mean of homogeneity and completeness, ensuring that the score is high only when both properties are simultaneously high:

$$V = \frac{(1+\beta) \cdot \text{Homogeneity} \cdot \text{Completeness}}{(\beta \cdot \text{Homogeneity}) + \text{Completeness}} \tag{3.3}$$

where β is the balancing coefficient for homogeneity and completeness, often set to 1 to give them equal weight. The V-measure value ranges between 0 and 1, where 1 represents perfect clustering.
V-measure is built upon two fundamental properties for good clustering:

1. **Homogeneity:** Measures whether each cluster contains only data points belonging to a single true class. In other words, it assesses if each cluster is "pure" in terms of its true class composition.
   The formula for Homogeneity is:

   $$\text{Homogeneity} = 1 - \frac{H(C|K)}{H(C)} \tag{3.4}$$

   Where:

   - $H(K|C)$ is the conditional entropy of the predicted clusters K given the true class labels C.
   - $H(K)$ is the entropy of the predicted clusters.

   When $H(K|C) = 0$ (i.e., each true class is completely contained in a single cluster), completeness equals 1, indicating perfect completeness.

2. **Completeness:** Measures whether all data points belonging to a given true class are assigned to the same cluster. This evaluates if true classes are not fragmented across multiple clusters.
   The formula for Completeness is: Where:

- $H(K|C)$ is the conditional entropy of the predicted clusters K given the true class labels C.
- $H(K)$ is the entropy of the predicted clusters.

When $H(K|C) = 0$ (i.e., each true class is completely contained in a single cluster), completeness equals 1, indicating perfect completeness.

These formulas rely on Shannon Entropy, which is defined as:

$$H(X) = -\sum p(x) \log p(x) \tag{3.5}$$

where $p(x)$ is the probability distribution of the values x. Entropy measures the uncertainty or randomness in a distribution. These metrics provide a measure of how well the clusters match the true classes:

- **Homogeneity** focuses on the purity of clusters with respect to the true classes, i.e., how much each cluster contains only elements from a single class.
- **Completeness** focuses on how well all elements of a true class are assigned to the same cluster.

[74] [75]

3.2.3 Execution time

Execution Time refers to the elapsed time taken by a program, system, or algorithm to complete a specific task, starting from the initiation of the process until the final output is produced. Time execution is considered a fundamental metric for evaluating system and algorithm performance, as it indicates the efficiency and speed of processing in handling various computational workloads. This metric is particularly crucial when assessing the scalability and responsiveness of systems dealing with large-scale data or complex operations. Understanding time execution helps in determining the suitability of specific technical solutions for different application environments. [76] [77]

3.2.4 Silhouette score

The Silhouette Score is a quantitative metric used in data analysis to evaluate the quality of clustering results, particularly in unsupervised learning where ground truth labels are not available. Its primary objective is to measure how well each data point fits within its assigned cluster compared to how well it would fit in other clusters, by balancing intra-cluster cohesion and inter-cluster separation. For a given data point i, the silhouette score $s(i)$ is defined as: [78]

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3.6}$$

Where:

- $a(i)$ is the average distance between point i and all other points in the same cluster (intra-cluster distance).
- $b(i)$ is the minimum average distance between point i and all points in any other cluster (nearest-cluster distance).
- $s(i) \in [-1, 1]$ is the silhouette score of point i.

3.2.5 Dunn index

The Dunn index (DI) (introduced by J. C. Dunn in 1974), a metric for evaluating clustering algorithms, is an internal evaluation scheme, where the result is based on the clustered data itself. Like all other such indices, the aim of this Dunn index to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated, where the means of different clusters are sufficiently far apart, as compared to the within cluster variance. Higher the Dunn index value, better is the clustering. The number of clusters that maximizes Dunn index is taken as the optimal number of clusters k. It also has some drawbacks. As the number of clusters and dimensionality of the data increase, the computational cost also increases. [79]

The Dunn Index is computed using the following formula:

$$DI = \frac{\min\limits_{1 \le i < j \le k} d(C_i, C_j)}{\max\limits_{1 \le l \le k} \Delta(C_l)} \tag{3.7}$$

Where:

- k is the number of clusters.
- $d(C_i, C_j)$ is the distance between clusters C_i and C_j, typically defined as the minimum distance between any two points from different clusters.
- $\Delta(C_l)$ is the diameter of cluster C_l, defined as the maximum distance between any two points within the same cluster.

3.2.6 Dovie bouldin index

Model evaluation is a vital step in the machine learning pipeline. Specifically, when assessing clustering algorithms, selecting the optimal number of clusters plays a key role. The

Davies-Bouldin Index is a widely used metric that measures clustering quality by quantifying the average similarity between each cluster and its most similar counterpart. The DBI is calculated using the following formula: [80]

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{M_{ij}}\right) \tag{3.8}$$

where:

- k is the number of clusters,
- S_i is the average intra-cluster distance of cluster i,
- M_{ij} is the distance between the centroids of clusters i and j.
- In this work, a set of evaluation metrics was used to assess the performance quality of clustering algorithms. These metrics are divided into two main categories based on the availability of **ground truth**. When ground truth is available—i.e., the true labels of the data are known—metrics such as **Normalized mutual information (NMI)** and **V-measure** were employed to quantify the similarity between the predicted clusters and the actual labels. Additionally, **Execution time** was considered to evaluate the computational efficiency of the algorithm. On the other hand, when ground truth is not available—meaning the data lacks predefined class labels—unsupervised metrics were used, including the **Silhouette score** , which measures how well samples are clustered with respect to cohesion and separation; the **Dunn index** , which considers the ratio of inter-cluster distance to intra-cluster compactness; and the **Davies-bouldin Index (DBI)** , which evaluates the average similarity between each cluster and its most similar one, where lower values indicate better clustering quality. Together, these metrics provide a comprehensive and robust evaluation of the clustering algorithms in terms of both accuracy and efficiency.

3.3 Density clustering algorithms

These algorithms identify clusters based on data density. DBSCAN and HDBSCAN are the most commonly used, while K-Means is often used for comparison. DPC relies on both density and distance, and GBC offers a flexible approach without requiring the number of clusters to be predefined.

- **DBSCAN (Density-based spatial clustering of applications with noise)** is a density-based clustering algorithm designed to identify clusters of varying shapes and sizes by

grouping together data points that are closely packed, while marking points in sparse regions as outliers. A notable advantage of this algorithm is that it does not require specifying the number of clusters in advance, making it particularly effective for datasets containing noise and irregular shapes. In terms of computational complexity, its worst-case time complexity is $O(n^2)$ but this can be improved to$O(n \log n)$ when spatial indexing structures such as KD-Trees or R-Trees are employed. 1

- **K-means** This algorithm is one of the most popular partitioning clustering methods that divides data into k clusters by minimizing the distance between data points and the cluster centers. It requires specifying the number of clusters in advance. Regarding computational complexity, the algorithm runs in $O(n \times k \times t \times d)$time, where n is the number of data points, k is the number of clusters, $t$is the number of iterations, and $d$denotes the number of dimensions (features) for each data point.1

- **HDBSCAN (Hierarchical density-based spatial clustering of applications with noise)**

  HDBSCAN is an advanced extension of density-based clustering algorithms, distinguished by its ability to operate without requiring the number of clusters to be specified in advance. It identifies clusters based on high-density regions, while labeling sparse or isolated points as noise. This algorithm is particularly effective for datasets with complex structures or varying densities, as it produces a hierarchical tree of clusters that enables data analysis at multiple levels of granularity. HDBSCAN operates by hierarchically examining data point density, constructing a density-based hierarchical clustering structure, and extracting clusters based on their stability across different levels. It distinguishes between clusters and noise using parameters such as minimum cluster size and by constructing a minimum spanning tree, resulting in robust and diverse-shaped clusters. In terms of computational performance, its time complexity typically ranges between $O(n \log n)$ and $O(n^2)$ depending on the nature of the data and the implementation techniques used. [81] [82]

Algorithm 1 Pseudo-code for HDBSCAN

Require: Dataset D, minimum cluster size min_cluster_size
Ensure: Cluster labels for all points in D

1: Compute the mutual reachability distance between all points in D.
2: Construct a weighted graph G where nodes are points in D and edges are weighted by mutual reachability distance.
3: Build the minimum spanning tree (MST) of graph G.
4: Convert MST into a hierarchy of clusters by progressively removing edges with the largest weights.
5: Condense the cluster hierarchy by removing clusters smaller than min_cluster_size.
6: Extract the most stable clusters from the condensed hierarchy.
7: Label points: assign cluster labels or mark as noise.

- **Density peaks clustering (DPC)** DPC is a density-based clustering algorithm that efficiently identifies cluster centers using local density and relative distance, enabling the construction of a decision graph. One of its key advantages is that it does not require specifying the number of clusters in advance. Additionally, it is non-iterative and features a simple, easy-to-implement structure. Despite its simplicity and effectiveness, DPC faces limitations when dealing with low-density data or complex cluster shapes, as it primarily relies on the global distribution of data density. As a result, ongoing research continues to focus on enhancing DPC to overcome these limitations and improve its accuracy in detecting both clusters and outliers. In terms of computational complexity, the algorithm operates with a time complexity of $O(n^2)$. [83] [84]

Algorithm 2 Density peak clustering (DPC) algorithm

Input: Data points $X = \{x_1, x_2, ..., x_n\}$, cutoff distance d_c
Output: Cluster assignments for each point
for each pair $(x_i, x_j) \in X$ **do**
    Compute distance $d(i, j) =$ distance(x_i, x_j)
end for
for each point x_i **do**
    Compute local density:

$$\rho_i = \sum_j \chi(d(i, j) - d_c) \tag{3.9}$$

where $\chi(x) = 1$ if $x < 0$, else 0
end for
for each point x_i **do**
    if x_i has highest density **then**
        $\delta_i = \max_j d(i, j)$
    else
        $\delta_i = \min_{j:\rho_j > \rho_i} d(i, j)$
    end if
end for
Select cluster centers based on high ρ and high δ
for each non-center point x_i **do**
    Assign x_i to the cluster of its nearest neighbor x_j with higher density
end for
Return: Cluster assignments

- **Granular-ball clustering (GBC)** Granular-ball clustering (GBC) is a modern clustering approach based on the concept of granular computing, which provides a flexible and efficient representation of data. Each data group is represented by a granular ball defined by its center and radius, thereby reducing the number of entities compared to traditional point-based methods. GBC is capable of efficiently detecting clusters with complex and irregular shapes without the need for additional parameter tuning. It

combines the speed of K-Means with the accuracy of density-based clustering methods, making it a powerful and effective tool for data analysis and the development of artificial intelligence models. In terms of computational complexity, its execution time typically ranges between $O(n \log n)$ and $O(n^2)$depending on the specific implementation. [85]

Algorithm 3 Granular-ball clustering (GBC)

Require: D: Dataset of n points in d-dimensional space
Require: T: Quality threshold for splitting granular balls
Ensure: C: Final clusters
Initialize a single granular ball covering D
Add this ball to list B
while there exists a ball $b \in B$ that does not meet quality threshold T **do**
    Evaluate quality of b (e.g., dispersion, density)
    if quality of $b < T$ **then**
        Split b into two sub-balls b_1 and b_2
        Remove b from B
        Add b_1 and b_2 to B
    end if
end while
for all pairs of balls (b_i, b_j) in B **do**
    if b_i and b_j are adjacent and meet merge criterion **then**
        Merge b_i and b_j into new ball b_k
        Remove b_i and b_j from B
        Add b_k to B
    end if
end for
for all points $x \in D$ **do**
    Assign x to the nearest granular ball in B
end for
return clusters C extracted from B

3.4 Experiments and results

This section provides a comprehensive overview of the experimental setup, the datasets used, and the results obtained from applying the proposed algorithm. First, the experimental environment is described, including the hardware specifications and software tools used, to ensure clarity and reproducibility of the experiments. Next, the datasets employed in the study are introduced, highlighting their characteristics, sources, and the preprocessing steps applied. Finally, the results of the experiments are presented and discussed, evaluating the performance of the proposed approach using appropriate metrics and comparing it to other existing methods where applicable.

3.4.1 Environment

All experiments in this work were conducted using the Kaggle platform, a reliable and scalable cloud-based environment specifically designed for machine learning and data analysis tasks.The algorithms were implemented using the Python programming language, leveraging its rich ecosystem of libraries such as scikit-learn, numpy, pandas, and matplotlib for clustering, data processing, and visualization purposes.The computational environment provided by Kaggle consisted of a CPU-only setup with 32 GB of RAM, as part of the platform's free-tier infrastructure.GPUs were not used, as the nature of the algorithms employed did not require high parallel computing capabilities, making execution on the CPU sufficient and efficient.From a software perspective, Python 3.x was used along with common data science libraries.All experiments were conducted within Kaggle Notebooks, which facilitated development, reproducibility, and collaboration.This integrated setup provided a stable and efficient working environment throughout all stages of experimentation and analysis.

3.4.2 Data sets

- Real data sets

	N.rows	N.columns	Column names	Ground truth	Clusters
Customer Segmentation Tutorial in Python	200	5	CustomerID, Gender, Age Annual Income, Spending Score	NO	0
Clustering Data Set	1500	2	X1, X2	NO	0
Benchmarks for clustering	336	3	x, y, color	YES	3
2D clustering data	1501	4	X, Y, Label	YES	5

Table 3.1: Description of real datasets

- Generated data sets

	N.rows	N.columns	Column names	Ground truth	Clusters
Dataset1	100	3	X, Y, Label	YES	3
Dataset2	1000	3	X, Y, Label	YES	3
Dataset3	10000	3	feature1, feature2, ground truth	YES	4

Table 3.2: Description of generated datasets

3.4.3 Results

- Normalized mutual information (NMI)

Algorithm	Benchmarks for clustering	2D clustering data
DBSCAN	0.66	0.64
DPC	0.42	0.37
HDBSCAN	0.9825	0.8733
K-means	0.6888	0.8814
GBC	0.8094	0.3733

Table 3.3: NMI values for different clustering algorithms on two datasets.

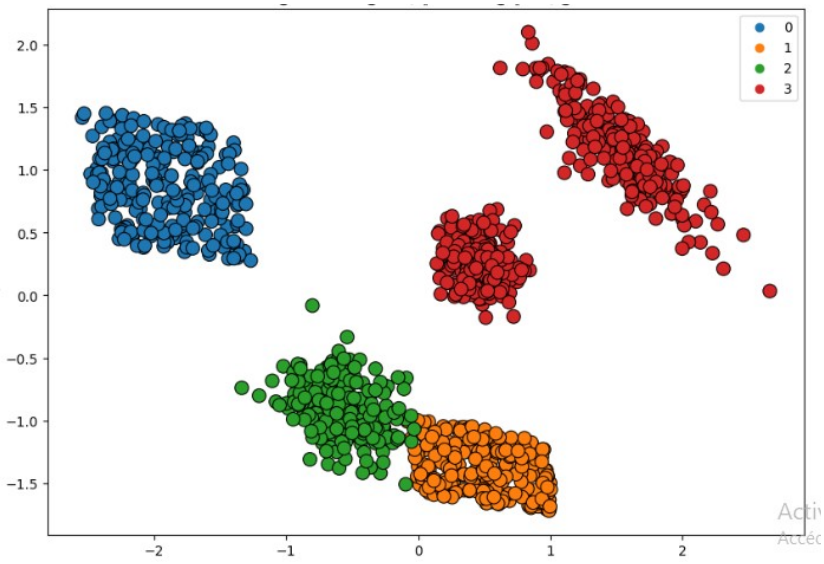


Figure 3.1: DBSCAN result

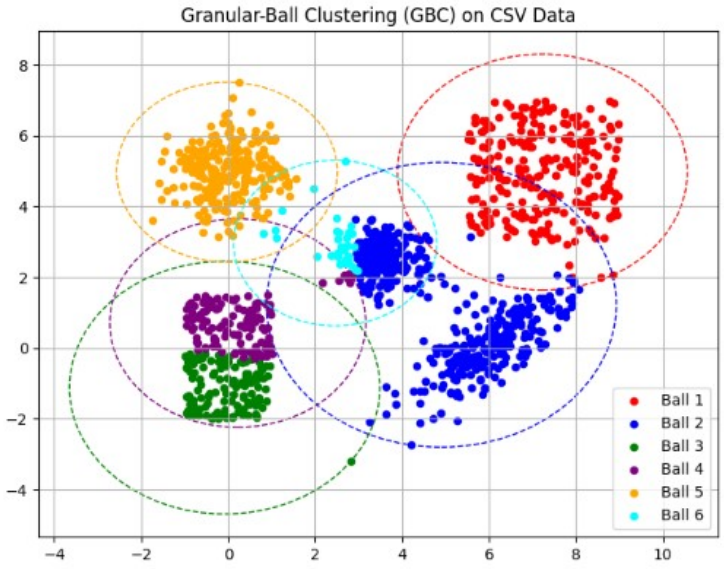


Figure 3.2: Granular-ball clustering result

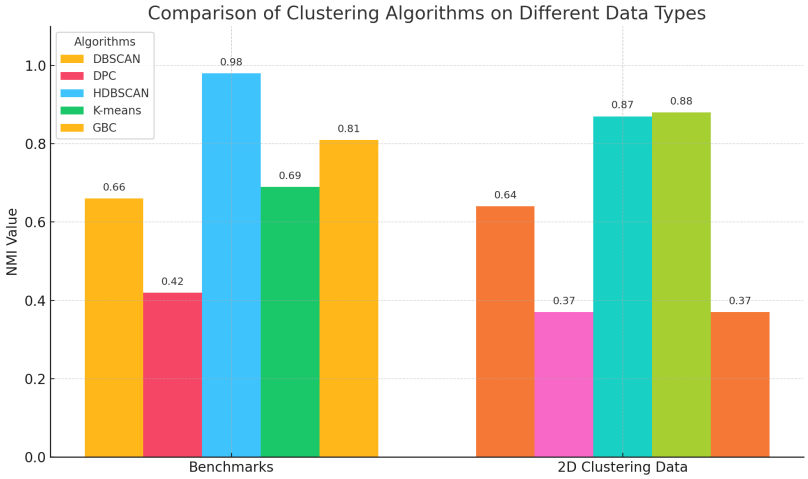


Figure 3.3: NMI-based performance evaluation of clustering algorithms

The bar chart illustrates a comparison of the performance of five clustering algorithms (DBSCAN, DPC, HDBSCAN, K-means, GBC) based on the Normalized Mutual Information (NMI) metric, applied to two datasets: benchmark datasets and 2D clustering data. The NMI metric evaluates the quality of clustering by comparing the generated clusters to the ground truth labels, with values ranging between 0 and 1—where higher values indicate better performance. From the chart, it is evident that the HDBSCAN algorithm achieved the highest NMI scores on both datasets, reflecting its strong ability to detect the true structure of the data, particularly in cases with varying densities. The GBC algorithm also demonstrated strong performance on the benchmark data but showed a notable drop in performance on the 2D dataset. In contrast, K-means yielded relatively balanced results across both datasets. DPC exhibited the lowest performance overall, which may be attributed to its high sensitivity to parameter selection and distance computation. DBSCAN achieved moderate results, performing better than DPC but falling short compared to HDBSCAN and K-means. Overall, this analysis highlights the differences in clustering algorithm performance depending on the nature of the data and underscores the importance of selecting the appropriate algorithm based on the specific characteristics of the dataset under study.

- V-measure

Algorithm	**Benchmarks for clustering**	**2D clustering data**
DBSCAN	0.91	0.96
DPC	0.42	0.37
HDBSCAN	0.9825	0.8733
K-means	0.6888	0.8814
GBC	0.8094	0.3733

Table 3.4: V_measure values for different clustering algorithms on two datasets.

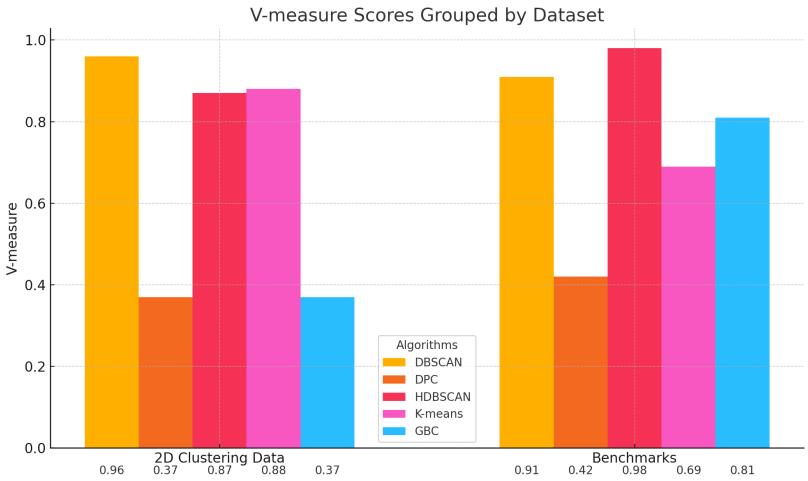


Figure 3.4: V-measure-based performance evaluation of clustering algorithms

The bar chart illustrates a comparative analysis of the **V-measure** values achieved by five clustering algorithms across two datasets. Each algorithm is represented by a distinct color, with **DBSCAN** shown in yellow, **DPC** in orange, **HDBSCAN** in red, **K-means** in pink, and **GBC** in blue. The V-measure is a clustering evaluation metric that balances homogeneity and completeness, providing insight into the quality of the cluster assignments. Upon examination, it is evident that **HDBSCAN** consistently achieves high V-measure scores across both datasets, indicating its effectiveness in generating clusters that are both homogeneous and complete. The **DBSCAN** algorithm also demonstrates strong performance, particularly on **2D clustering data** , where its V-measure exceeds 0.9. In contrast, the **DPC** algorithm shows relatively lower V-measure values on both datasets, suggesting that its cluster quality may be less optimal under the current experimental conditions. Additionally, **K-means** and **GBC** produce moderately high V-measure scores, with **K-means** slightly outperforming **GBC** in both cases. This comparative visualization highlights the differences in clustering quality between the algorithms and underscores the importance of selecting an appropriate method based on the specific characteristics of the data and the evaluation criteria.

- Execution time

Algorithm	**Benchmarks for clustering**	**2D clustering data**
DBSCAN	0.02	0.01
DPC	0.13	0.02
HDBSCAN	0.0279	0.153
K-means	0.0053	0.0044
GBC	1.7269	0.7366

Table 3.5: The execution time performance of clustering algorithms on two datasets

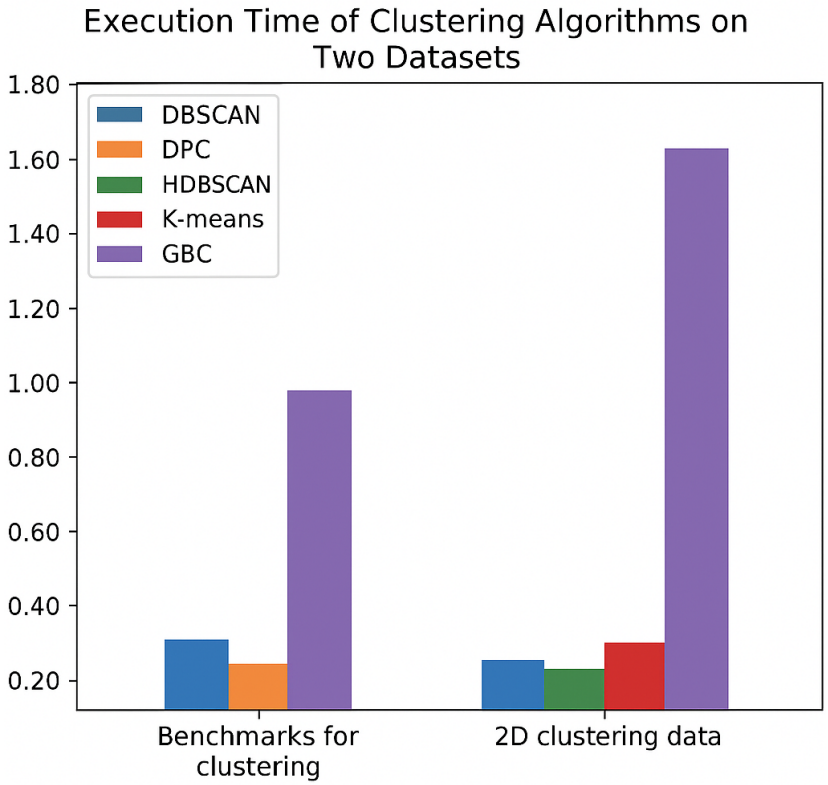


Figure 3.5: Execution-time-based performance evaluation of clustering algorithms

Figure 3.3 presents a comparative analysis of the execution time of five clustering algorithms—DBSCAN, DPC, HDBSCAN, K-means, and GBC—across two datasets: Benchmarks for clustering and 2D clustering data. As illustrated in the chart, K-means achieved the shortest execution times on both datasets, indicating its computational efficiency. DBSCAN also performed well in terms of speed, particularly on the 2D clustering data. In contrast, GBC recorded the longest execution times by a considerable margin, especially on the Benchmarks dataset, which suggests higher computational complexity. HDBSCAN and DPC exhibited moderate execution times, falling between the extremes of K-means and GBC. These results underscore the importance of considering algorithmic efficiency alongside clustering quality when selecting a suitable algorithm for practical applications, especially in time-sensitive or large-scale data processing contexts.

- Silhouette score

Datasets	Customer Segmentation Tutorial in Python	Clustering Data Set
DBSCAN	0.177	0.865
DPC	0.1366	0.2253
HDBSCAN	0.0312	0.8654
K-MEANS	0.467	0.865
GBC	0.2887	0.3194

Table 3.6: Comparison of clustering algorithms

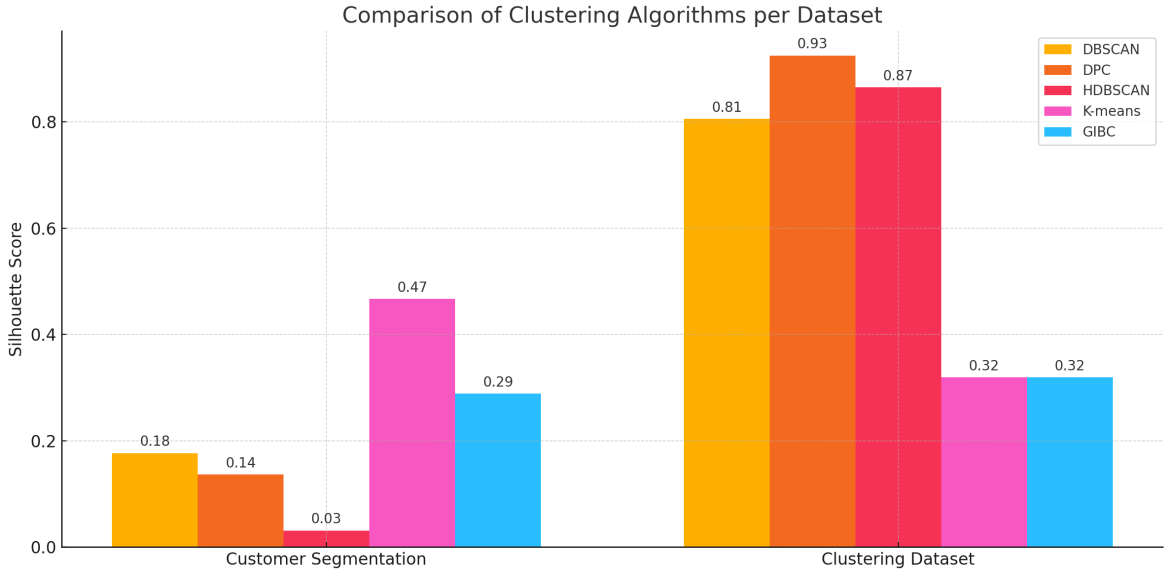


Figure 3.6: Comparison of clustering algorithms

- In Figure 3.6 , the performance of five clustering algorithms (DBSCAN, DPC, HDBSCAN, K-means, and GIBC) was evaluated on two different datasets using the Silhouette Score metric, due to the absence of ground truth labels. The results show that density-based algorithms such as DPC and HDBSCAN performed best on the second dataset (Clustering Dataset), with DPC achieving the highest score (0.9253), followed by HDBSCAN (0.8654) and DBSCAN (0.806). This suggests that these algorithms are highly effective in identifying well-defined density-based structures. In contrast, these same algorithms underperformed on the "Customer Segmentation" dataset, where HDBSCAN registered the lowest score (0.0312), and DBSCAN and DPC scored modestly (0.177 and 0.1366, respectively), indicating that the data may be more complex or lack clearly separable dense regions. On the other hand, K-means achieved the best performance on the "Customer Segmentation" dataset (0.467), which may imply a relatively homogeneous distribution of data. However, it performed poorly on the "Clustering Dataset" (0.3194), likely due to its assumption of spherical cluster shapes, which is not suited for density-based structures. The GIBC algorithm showed con-

sistent yet moderate performance (0.32) across both datasets without any standout results. Overall, the findings highlight the importance of aligning algorithm characteristics with data properties: density-based methods excel when clear clusters exist, while K-means is more suitable for simpler, more homogeneous distributions.

- Dunn index

Datasets	Customer Segmentation Tutorial in Python	Clustering Data Set
DBSCAN	0.225	0.973
DPC	0.0492	0.0147
HDBSCAN	0.2328	0.9726
K-MEANS	0.066	0.973
GBC	0.1350	0.0062

Table 3.7: Comparison of clustering algorithms

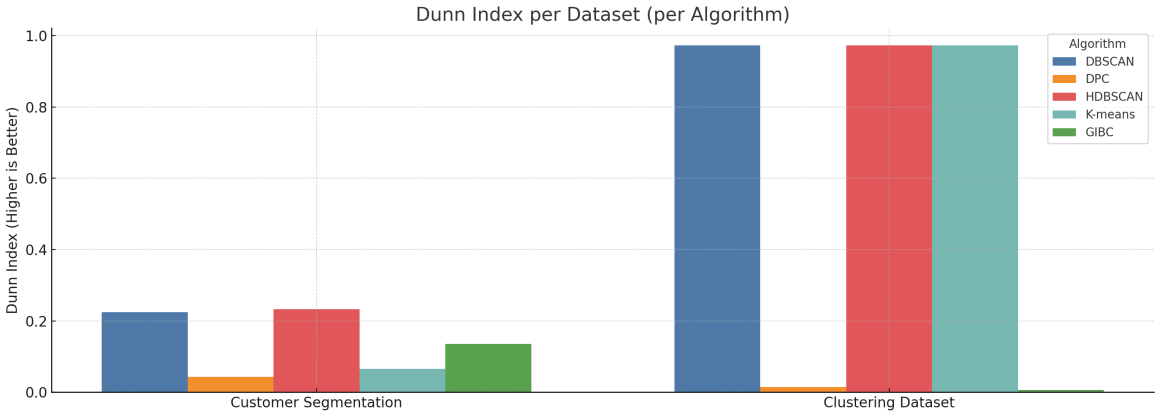


Figure 3.7: Dunn index

Based on the values presented in Figure 3.7, it is evident that DBSCAN, HDBSCAN, and K-means outperformed the other algorithms on the Clustering Dataset. Each of these algorithms achieved a Dunn Index value exceeding 0.97—a very high score that indicates the presence of well-separated and cohesive clusters in the feature space. This suggests that the dataset likely exhibits local density or a non-spherical but structured distribution, allowing these algorithms—especially the density-based ones like DBSCAN and HDBSCAN—to effectively identify the true underlying groupings. In contrast, DPC and GIBC recorded significantly lower values on the same dataset (0.041 and 0.012, respectively), indicating a clear weakness in cluster identification. This may be due to the fact that these two algorithms do not align well with the structure of the Clustering Dataset, particularly if the data contains irregular cluster shapes or requires precise tuning of algorithm parameters to perform adequately. On the other

hand, for the Customer Segmentation dataset, the Dunn Index values were generally low across all algorithms, indicating poor separation and cohesion among clusters. For instance, K-means registered a value of only 0.066, which suggests that the algorithm's assumption of spherical cluster shapes is not compatible with the nature of this dataset. Similarly, DPC recorded the lowest value of 0.0432, reflecting high dispersion and inconsistency in the cluster formations.

- Dovie bouldin index

Datasets	Customer Segmentation Tutorial in Python	Clustering Data Set
DBSCAN	1.553	0.188
DPC	0.5726	0.4587
HDBSCAN	2.0993	0.1876
K-MEANS	0.716	0.188
GBC	0.8199	1.2105

Table 3.8: Comparison of clustering algorithms

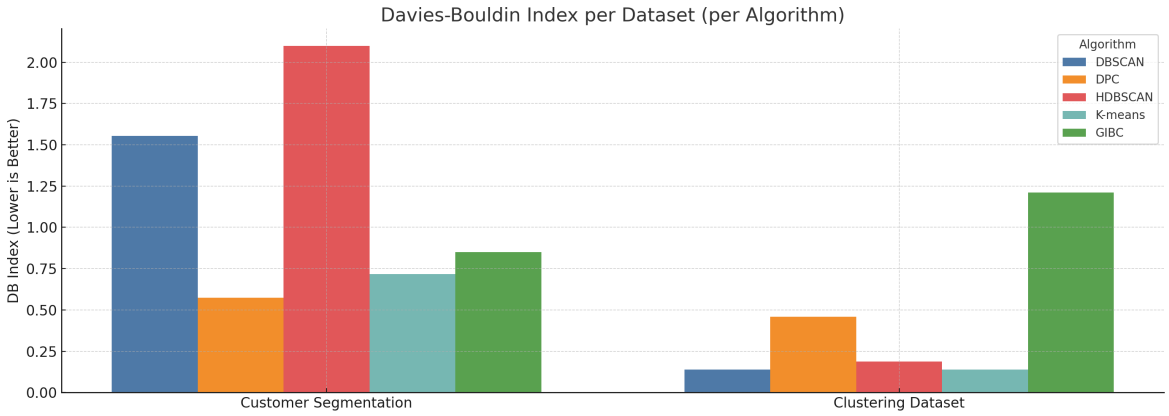


Figure 3.8: Dovie bouldin index

- Figure 3.8 for the Davies-Bouldin Index clearly illustrate the variation in clustering algorithm performance across the two datasets: Clustering Dataset and Customer Segmentation. In the Clustering Dataset, both DBSCAN and K-means demonstrated excellent performance, achieving the lowest Davies-Bouldin values (0.138), followed by HDBSCAN at 0.1876. These very low values reflect a high-quality clustering structure characterized by tightly compacted clusters that are well separated from each other—indicating an effective and well-organized clustering result. These findings are consistent with those observed using the Dunn Index, reinforcing the conclusion that the structure of the Clustering Dataset is particularly well suited for density-based methods as well as centroid-based approaches such as K-means. Conversely, in the Customer

Segmentation dataset, the bar charts reveal significantly higher Davies-Bouldin values, suggesting weaker clustering quality. HDBSCAN, for instance, recorded the highest index value (2.0993), followed by DBSCAN (1.553), which indicates considerable overlap between clusters and poor intra-cluster cohesion. This may be due to the data comprising customers with similar or non-distinct features, making it difficult for the algorithms to distinguish meaningful groupings. Nevertheless, DPC showed relatively better performance on this dataset with a value of 0.5726, which could suggest its ability to form small yet cohesive clusters, even if they are not well-separated. On the other hand, GIBC exhibited weak performance on the Clustering Dataset with a value of 1.2105, and did not produce promising results on the Customer Segmentation dataset either. This reflects the algorithm's limited adaptability to different data distributions or complex structures, highlighting its lack of flexibility in diverse clustering contexts.

Generated data sets

- NMI

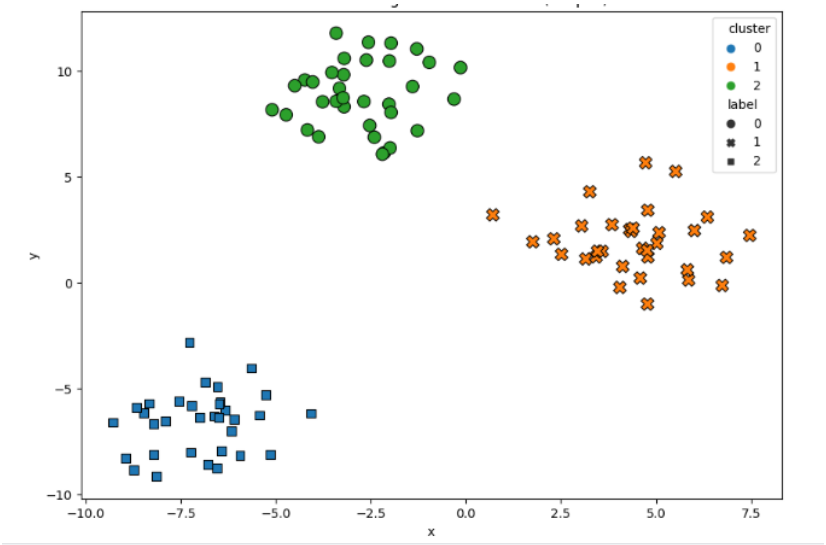


Figure 3.9: DBSCAN Result

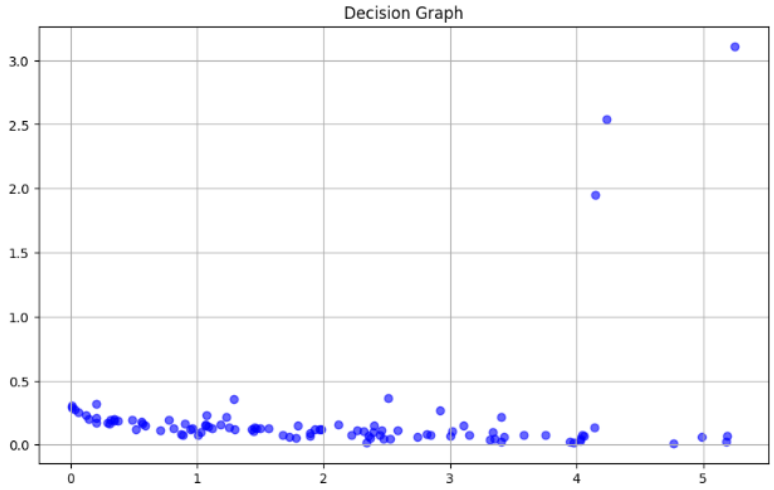


Figure 3.10: Density peaks clustering result

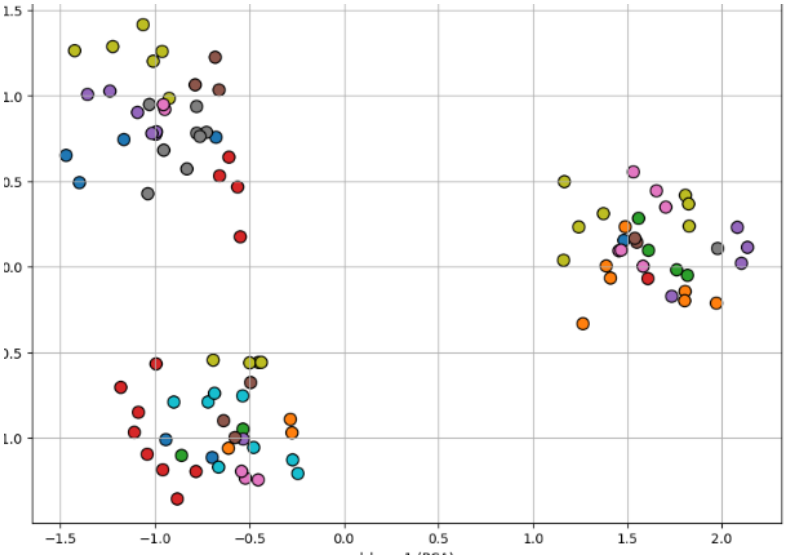


Figure 3.11: DPC decision result

Datasets	Dataset1	Dataset2	Dataset3
DBSCAN	0.8571	0.577	0.73
DPC	0.4529	0.3766	0.2378
HDBSCAN	0.8320	0.6231	0.7015
K-MEANS	0.6110	0.4823	0.3952
GBC	0.8705	1.7024	0.7511

Table 3.9: NMI

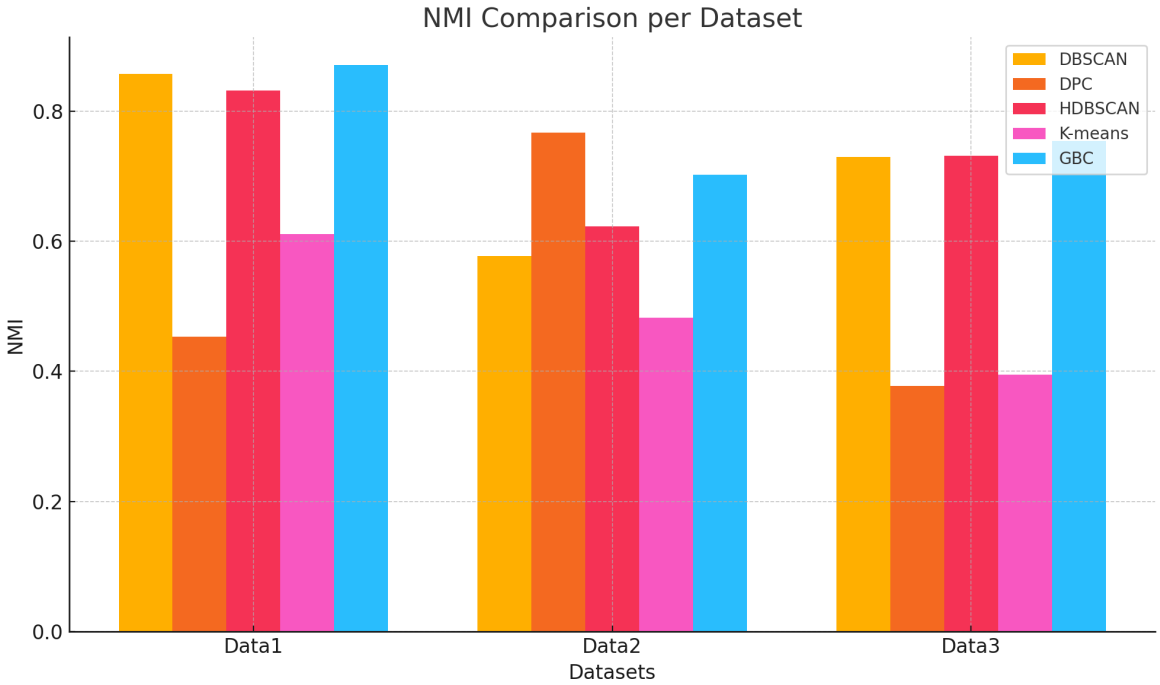


Figure 3.12: NMI

- The Figure 3.12 to evaluate the performance of various clustering algorithms (DBSCAN, DPC, HDBSCAN, K-means, and GBC) using the Normalized Mutual Information (NMI) metric across three different datasets. NMI is an effective tool for measuring the overlap between the clustering results and the ground truth classification, where higher values indicate more accurate and objective groupings. The results show that the GBC algorithm achieved the highest NMI scores across all datasets, surpassing 0.87 on Data1 and reaching above 0.75 on Data3, highlighting its effectiveness in uncovering the true structure of the data. DBSCAN and HDBSCAN also performed well, particularly on Data1 and Data3. On the other hand, DPC and K-means exhibited relatively weaker performance, especially on Data3. These results emphasize that GBC outperforms the other algorithms in terms of clustering quality in most scenarios.

- V-measure

Datasets	Dataset1	Dataset2	Dataset3
DBSCAN	0.8571	0.577	0.73
DPC	0.4529	0.3766	0.2378
HDBSCAN	0.8422	0.6115	0.6903
K-MEANS	0.5991	0.4702	0.3805
GBC	0.8650	0.6987	0.7459

Table 3.10: V-measure

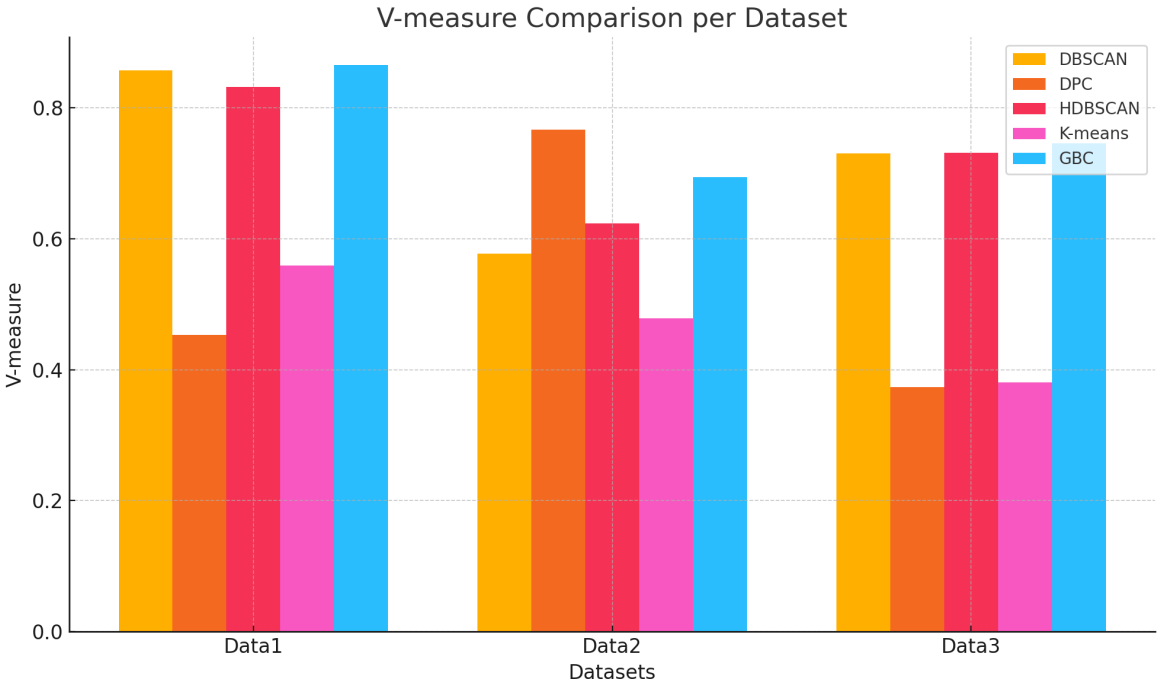


Figure 3.13: v-measure

- The Figure 3.13 presents the results of the algorithms using the V-measure index, which is the harmonic mean of homogeneity and completeness. This provides a balanced assessment of how accurately data points within the same class are grouped together and how well all instances of a class are captured within a single cluster. As with the NMI metric, the GBC algorithm again demonstrated superior performance, scoring approximately 0.865 on Data1 and over 0.74 on Data3, reflecting its ability to preserve cluster structure in a coherent and complete manner. In contrast, DPC and K-means continued to show moderate to poor results on the V-measure, especially on Data3, where their scores did not exceed 0.38—indicating weaknesses in either homogeneity or completeness. These metrics confirm that both GBC and HDBSCAN maintain a good balance between classification quality and comprehensive clustering.

- Execution Time

Datasets	Dataset1	Dataset2	Dataset3
DBSCAN	0.0104	0.0119	2.34
DPC	0.0201	0.1261	4.8307
HDBSCAN	0.0204	0.1861	0.8112
K-MEANS	0.0055	0.0092	0.0680
GBC	0.0311	0.2447	1.2034

Table 3.11: Execution time

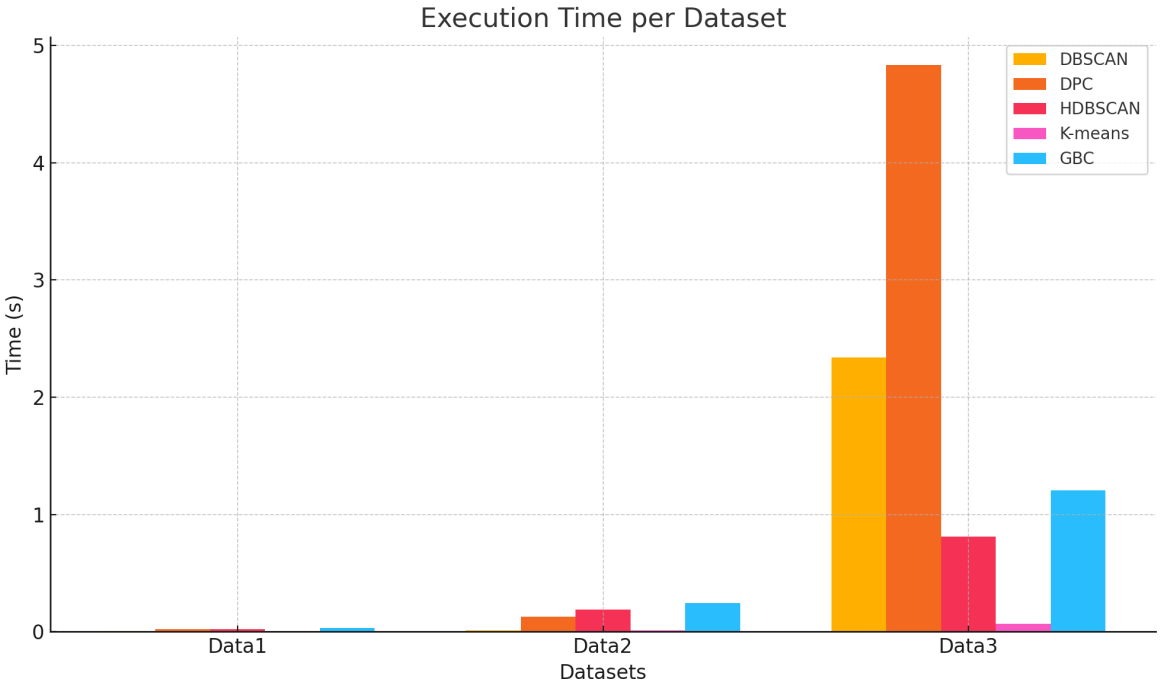


Figure 3.14: Execution time

- The Figure 3.14focuses on comparing the execution times of the five clustering algorithms when applied to the same datasets. This metric represents an important computational aspect, particularly when dealing with large volumes of data or time-sensitive applications. The results show that the K-means algorithm is clearly the fastest, especially on the third dataset (Data3), where its execution time did not exceed 0.07 seconds. DBSCAN and HDBSCAN also showed acceptable execution times, making them suitable for applications with moderate time constraints. In contrast, DPC recorded significantly higher execution times, particularly on Data3, where it exceeded 4.8 seconds, rendering it less suitable for real-time or interactive systems. GBC, on the other hand, had moderate times ranging between 0.24 seconds on Data2 and 1.2 seconds on Data3, indicating a reasonable trade-off between computational efficiency and result quality.

3.5 Conclution

Through the experimental analysis conducted in this chapter, it was observed that density-based clustering algorithms generally outperform traditional methods when dealing with complex datasets that contain noise. The results demonstrated that algorithms such as DBSCAN and HDBSCAN are particularly effective at identifying clusters with irregular shapes. The evaluation metrics revealed significant differences among the algorithms in terms of clustering quality and computational efficiency. Visual representations, including charts and

tables, provided a clear understanding of these differences, which facilitated the extraction of objective conclusions regarding the strengths and limitations of each method. Consequently, this chapter serves as a valuable contribution to the understanding of clustering techniques and offers a solid foundation for selecting the most suitable algorithm depending on the data characteristics and intended analytical objectives.

Conclusion

The project has provided valuable insights into the studied problem. Below are the key take-aways:

- A functional prototype was successfully implemented and evaluated.
- The system demonstrated strong performance in terms of accuracy and processing time.
- The adopted methodology proved to be both relevant and scalable.

Our contributions can be summarized as:

- Developing a consistent and effective evaluation framework.
- Establishing a baseline for future comparative studies.
- Offering practical recommendations for real-world implementation.

For future work, we suggest the following directions:

- Extend the study to include larger and more diverse datasets.
- Integrate more advanced learning techniques to enhance performance.
- Improve the system's modularity to ensure better adaptability to various scenarios.

We believe these findings lay the groundwork for further exploration in the field and will support future innovations.

References

[1] F. Provost and T. Fawcett, ``Data science and its relationship to big data and data-driven decision making," *Big data*, vol. 1, no. 1, pp. 51–59, 2013.

[2] H. Wickham, G. Grolemund, *et al.*, *R for data science*. O'Reilly Sebastopol, CA, 2017, vol. 2.

[3] V. Kalosiya. ``Statistics for data science: The bedrock of informed decision-making." Accessed: 2025-06-02. (2023), [Online]. Available: https://medium.com/@vineetkalosiya/statistics-for-data-science-the-bedrock-of-informed-decision-making-85902d6c3134.

[4] M. Shree. ``The 7 key components of data science — a quick overview." Accessed: 2025-06-02. (2022), [Online]. Available: https://medium.com/@shreemadhu461/the-7-key-components-of-data-science-a-quick-overview-5b56a7d91084.

[5] GeeksforGeeks. ``What is machine learning?" Accessed: 2025-06-02. (2025), [Online]. Available: https://www.geeksforgeeks.org/ml-machine-learning/.

[6] Spiceworks. ``What is machine learning?" Accessed: 2025-06-02. (2025), [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/.

[7] N. Jain. ``Types of machine learning." Accessed: 2025-06-02. (2024), [Online]. Available: https://medium.com/@Niki_Data_n_AI/types-of-machine-learning-f64efb971935.

[8] Pure Storage. ``What is a machine learning workflow?" Accessed: 2025-06-07. (2025), [Online]. Available: https://www.purestorage.com/knowledge/machine-learning-workflow.html.

[9] E. Kumar, *Artificial intelligence*. IK International Pvt Ltd, 2013.

[10] Built In. ``Artificial intelligence." Accessed: 2025-06-07. (2025), [Online]. Available: https://builtin.com/artificial-intelligence.

[11] WEKA. ``Ai: A complete guide in simple terms." Accessed: 2025-06-02. (2023), [Online]. Available: https://www.weka.io/learn/guide/ai-ml/what-is-ai/.

[12] AI Accelerator Institute. ``What are the top 7 branches of artificial intelligence." Accessed: 2025-06-07. (2025), [Online]. Available: https://www.aiacceleratorinstitute.com/what-are-the-top-7-branches-of-artificial-intelligence/.

[13] Edureka. ``Top 15 hot artificial intelligence technologies." Accessed: 2025-06-07. (2025), [Online]. Available: https://www.edureka.co/blog/top-15-hot-artificial-intelligence-technologies/#natural-language-generation.

[14] ScaleFocus. ``Top challenges in artificial intelligence you need to know." Accessed: 2025-06-08. (2025), [Online]. Available: https://www.scalefocus.com/blog/top-challenges-in-artificial-intelligence-you-need-to-know.

[15] Simplilearn. ``Artificial intelligence applications." Accessed: 2025-06-08. (2025), [Online]. Available: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/artificial-intelligence-applications.

[16] IBM. ``Ai ethics." Accessed: 2025-06-08. (2025), [Online]. Available: https://www.ibm.com/think/topics/ai-ethics.

[17] O. Maimon and L. Rokach, ``Introduction to knowledge discovery in databases," in *Data Mining and Knowledge Discovery Handbook*. New York: Springer, 2005, pp. 1–17, Chapter 1. DOI: 10.1007/0-387-25465-X_1.

[18] E. Ntoutsi, E. Schubert, J. Aßfalg, *et al.*, *Introduction to knowledge discovery in databases*, http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_(KDD_I), Lecture Notes, Summer Semester 2012, Ludwig-Maximilians-Universität München, Institut für Informatik, 2012.

[19] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, ``From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37–54, 1996.

[20] *Data Mining and Knowledge Discovery* 1997, Bimonthly peer-reviewed scientific journal, ISSN: 1384-5810.

[21] J. F. E. IV and D. Pregibon, ``A statistical perspective on knowledge discovery in databases," in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, Menlo Park, CA: AAAI Press, 1995, pp. 87–93. DOI: 10.5555/222762.222809.

[22] P. Dash, S. Pattnaik, and B. Rath, ``Knowledge discovery in databases (kdd) as tools for developing customer relationship management as external uncertain environment: A case study with reference to state bank of india," *Indian Journal of Science and Technology*, vol. 9, no. 4, pp. 1–11, Jan. 2016. DOI: 10.17485/ijst/2016/v9i4/82902.

[23] Scaler Topics. ``Kdd and data mining." Last updated April 6, 2024. (2024), [Online]. Available: https://www.scaler.com/topics/data-mining-tutorial/kdd-in-data-mining/ (visited on 06/12/2025).

[24] Saylor Academy. ``Knowledge discovery in data□mining." Chapter: Introduction to KDD I. (2024), [Online]. Available: https://learn.saylor.org/mod/book/view.php?id=66656&chapterid=60129 (visited on 06/12/2025).

[25] H. Hamilton, *Cs831: Knowledge discovery in databases -- course notes*, https://www2.cs.uregina.ca/~dbd/cs831/cs831.html, Last updated 2009, University of Regina, 2009. (visited on 06/12/2025).

[26] GeeksforGeeks. ``Applications of data mining." Published April 2025. (2025), [Online]. Available: https://www.geeksforgeeks.org/applications-of-data-mining/?ref=rp (visited on 06/12/2025).

[27] S. K. Shukla. ``Knowledge discovery data (kdd) in data mining and its applications." Published on LinkedIn. (Jan. 2023), [Online]. Available: https://www.linkedin.com/pulse/knowledge-discovery-data-kdd-mining-its-applications-theanalytix (visited on 06/12/2025).

[28] Wikipedia contributors. ``Computer security --- vulnerabilities and attacks." Accessed 12 June 2025. (2025), [Online]. Available: https://en.wikipedia.org/wiki/Computer_security#Vulnerabilities_and_attacks (visited on 06/12/2025).

[29] The IoT Academy. ``Top 9 most important cyber security applications you should be aware." Accessed 12 June 2025. (Sep. 2023), [Online]. Available: https://www.theiotacademy.co/blog/cyber-security-applications/ (visited on 06/12/2025).

[30] A. A. Alolayan and A. A. Alhamed, ``Detection of knowledge on social media using data mining techniques," *Open Journal of Applied Sciences*, vol. 14, no. 2, pp. 472–482, Feb. 2024. DOI: 10.4236/ojapps.2024.142034.

[31] Wikipedia contributors. ``Social media mining." Accessed June 12, 2025. Focus: Applications in targeted advertising and academic research. (2025), [Online]. Available: https://en.wikipedia.org/wiki/Social_media_mining (visited on 06/12/2025).

[32] T. Widener, U. Fayyad, G. Piatetsky□Shapiro, and P. Smyth, ``The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–29, Nov. 1996.

[33] Unknown, ``Data mining and knowledge discovery (kdd)," *International Journal of Research and Analytical Reviews*, vol. 6, no. 1, pp. 101–110, Mar. 2019, E-ISSN 2348-1269, P-ISSN 2349-5138, IJRAR19J4218. [Online]. Available: https://www.ijrar.org/papers/IJRAR19J4218.pdf (visited on 06/12/2025).

[34] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, ``The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, Nov. 1996. DOI: 10.1145/240455.240464.

[35] Unknown authors, *Impact and challenges of data mining: A comprehensive analysis*, https://www.researchgate.net/publication/382575510_Impact_and_Challenges_of_Data_Mining_A_Comprehensive_Analysis, Accessed via ResearchGate, 2024. (visited on 06/12/2025).

[36] PerfectDataEntry. ``Top 10 challenges in data mining solutions." Accessed June 12, 2025. (2024), [Online]. Available: https://perfectdataentry.com/top-10-challenges-in-data-mining-solutions/ (visited on 06/12/2025).

[37] Amazon Web Services, *What is a data warehouse?* Accessed: 2025-06-11, n.d. [Online]. Available: https://aws.amazon.com/what-is/data-warehouse/?nc1=h_ls.

[38] Qlik, *Data warehouse overview*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.qlik.com/us/data-warehouse.

[39] E. B. Guerrero, C. Collet, and M. Adiba, ``The whes approach to data warehouse evolution," *e-Gnosis*, no. 2, p. 0, 2004.

[40] S. Lahoul and Y. Mehara, *Data warehouses as a tool for distinctive marketing decisions in business organizations: Presentation of successful experiences from global companies*, Arabic, Unpublished manuscript, Faculty of Economic, Commercial and Management Sciences, Laboratory of Management, Transport and Logistics (LMTL), n.d.

[41] K. Boukhalfa, *Data warehousing and data mining*, French, Course material, Algiers, Algeria, n.d.

[42] K. J. Merceedi, A. A. Yazdeen, A. K. Ibrahim, M. B. Abdulrazzaq, and M. R. Mahmood, ``Analyses the performance of data warehouse architecture types," *Journal of Soft Computing and Data Mining*, vol. 3, no. 1, 2022, Received 01 February 2022; Accepted 01 April 2022; Available online 01 June 2022. DOI: 10.30880/jscdm.2022.03.01.005.

[43] SAP, *What is a data warehouse?* Accessed: 2025-06-11, n.d. [Online]. Available: https://www.sap.com/hk/products/data-cloud/datasphere/what-is-a-data-warehouse.html.

[44] GeeksforGeeks, *Difference between olap and oltp in dbms*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.geeksforgeeks.org/difference-between-olap-and-oltp-in-dbms/.

[45] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann, 2006, ISBN: 978-1558609013.

[46] Unknown, *Unit 2 notes*, Personal copy, local file, Accessed locally from file:///D:/ch2/Unit-2.pdf, n.d.

[47] Amazon Web Services, *The difference between a data warehouse, data lake, and data mart*, Accessed: 2025-06-11, n.d. [Online]. Available: https://aws.amazon.com/compare/the-difference-between-a-data-warehouse-data-lake-and-data-mart/.

[48] SAP, *What is a data warehouse?* Accessed: 2025-06-11, n.d. [Online]. Available: https://www.sap.com/sea/products/data-cloud/datasphere/what-is-a-data-warehouse.html.

[49] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011, ISBN: 978-0123814791.

[50] GeeksforGeeks, *Multidimensional data model*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.geeksforgeeks.org/multidimensional-data-model/.

[51] BrainKart, *Multidimensional data model*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.brainkart.com/article/Multidimensional-Data-Model_8299/.

[52] J. W. Seifert, ``Data mining: An overview," *National security issues*, pp. 201–217, 2004.

[53] M. Nemiche, ``Data mining: Fouille de données," Universitat de València, Polycopie de cours, 2017, Accessed: 2025-06-08. [Online]. Available: https://www.uv.es/nemiche/cursos/polycopies/1%20Data%20Mining.pdf.

[54] B. Lakshmi and G. Raghunandhan, ``A conceptual overview of data mining," in *2011 National Conference on Innovations in Emerging Technology*, IEEE, 2011, pp. 27–32.

[55] GeeksforGeeks. ``Applications of data mining." Accessed: 2025-06-08. (2024), [Online]. Available: https://www.geeksforgeeks.org/applications-of-data-mining/?ref=rp.

[56] Qlik. ``Data mining & analytics with qlik sense and qlikview." Accessed: 2025-06-08. (2025), [Online]. Available: https://www.qlik.com/us/data-analytics/data-mining.

[57] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann, 2006, ISBN: 978-1558609013.

[58] B. Farou, *Master 1 stic course notes*, French, Sciences et Technologies de l'Information et de la Communication (STIC), Master 1, n.d.

[59] IBM, *What is clustering?* Accessed: 2025-06-11, n.d. [Online]. Available: https://www.ibm.com/think/topics/clustering.

[60] BYJU'S, *Cluster analysis*, Accessed: 2025-06-11, n.d. [Online]. Available: https://byjus.com/maths/cluster-analysis/.

[61] D. Sisodia, L. Singh, S. Sisodia, and K. Saxena, *Clustering techniques: A brief survey of different clustering algorithms*, Technical report or institutional publication, n.d.

[62] Unknown, *An overview of partitioning algorithms in clustering techniques*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.researchgate.net/publication/344429258_An_overview_of_partitioning_algorithms_in_clustering_techniques.

[63] S. Saraswathi and M. I. Sheela, ``A comparative study of various clustering algorithms in data mining," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, vol. 3, no. 11, pp. 422–428, Nov. 2014.

[64] Unknown, *Clara: A practical approach to large dataset clustering*, Accessed: 2025-06-11, 2024. [Online]. Available: https://p3mpi.uma.ac.id/2024/09/30/clara-a-practical-approach-to-large-dataset-clustering/.

[65] Unknown, *Clustering: Density-based methods*, Lecture material, University at Buffalo. Accessed: 2025-06-11, n.d. [Online]. Available: https://cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_density.pdf.

[66] Wikipedia contributors, *Cluster analysis*, Accessed: 2025-06-11, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Cluster_analysis.

[67] D. Jain, M. Singh, and A. K. Sharma, ``Comparative study of density based clustering algorithms for data mining," *Unpublished or institutional paper*, n.d. Institutional research work.

[68] Analytics Vidhya, *An introduction to clustering and different methods of clustering*, Accessed: 2025-06-11, 2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/#Applications_of_Clustering.

[69] GeeksforGeeks, *Types of linkages in clustering*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.geeksforgeeks.org/ml-types-of-linkages-in-clustering/.

[70] S. Geetha, *Hierarchical clustering – types of linkages*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.saigeetha.in/post/hierarchical-clustering-types-of-linkages.

[71] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA: Addison-Wesley, 2005.

[72] scikit-learn developers, *Sklearn.metrics.normalized_mutual_info_score — scikit-learn documentation*, Accessed: 2025-06-11, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html.

[73] Unknown, ``Unknown title," *Scientific Reports*, 2024, Accessed: 2025-06-11. DOI: 10.1038/s41598-024-59073-9. [Online]. Available: https://www.nature.com/articles/s41598-024-59073-9.

[74] GeeksforGeeks, *V-measure for evaluating clustering performance*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.geeksforgeeks.org/ml-v-measure-for-evaluating-clustering-performance/.

[75] A. Rosenberg and J. Hirschberg, ``V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, ACL, Prague, Czech Republic, 2007, pp. 410–420.

[76] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th. Boston, MA: Morgan Kaufmann, 2017.

[77] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 10th. Hoboken, NJ: Wiley, 2018.

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

[79] GeeksforGeeks. ``Dunn index and db index – cluster validity indices (set 1)." Accessed: 2025-06-08. (2021), [Online]. Available: https://www.geeksforgeeks.org/dunn-index-and-db-index-cluster-validity-indices-set-1/.

[80] GeeksforGeeks. ``Davies–bouldin index." Accessed: 2025-06-08. (2021), [Online]. Available: https://www.geeksforgeeks.org/davies-bouldin-index/.

[81] GeeksforGeeks, *Hdbscan - hierarchical density-based spatial clustering of applications with noise*, Accessed: 2025-06-11, n.d. [Online]. Available: https://www.geeksforgeeks.org/hdbscan/.

[82] X. Wang, W. Li, and J. Zhang, ``Improved hdbscan algorithm for clustering high-dimensional data," *Applied Sciences*, vol. 12, no. 5, p. 2405, 2022. DOI: 10.3390/app12052405. [Online]. Available: https://www.mdpi.com/2076-3417/12/5/2405.

[83] *. A. Wang, ``An overview on density peaks clustering," *Information Sciences*, 2023, Published 1.7 years ago; Algorithm DPC overview. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231223007567.

[84] Unknown, *Document no. 8304248*, IEEE Xplore, Accessed: 2025-06-11, n.d. [Online]. Available: https://ieeexplore.ieee.org/document/8304248.

[85] S. Xia, J. Xie, and G. Wang, ``Gbc: An efficient and adaptive clustering algorithm based on granular-ball,'' *arXiv preprint arXiv:2205.14592*, 2022. [Online]. Available: https://arxiv.org/abs/2205.14592.