People's Democratic Republic of Algeria

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University of 8 May 1945 – Guelma -

Faculty of Mathematics, Computer Science and Material Sciences

Computer Science Department

Master's Thesis

Speciality: Computer Science

Option : Information Systems

Theme

A Comparative Study of Data Mining Tools

Presented by :

FETATNIA Abdellah

Jury Members:

N	**Full Name**	**Quality**
1	Dr. FAREK Lazhar	**Chairman**
2	Dr. AGGOUNE Aicha	**Supervisor**
3	Dr. FARKOUS Chokri	**Examiner**

June 25, 2024

Acknowledgments

First and foremost, I thank Almighty God for granting me the patience, courage, and determination to succeed in my theses.

I would like to extend my heartfelt gratitude to everyone who has supported and guided me throughout the course of my master's studies and the completion of this thesis.

My deepest appreciation goes to my supervisor, Dr. Aggoune Aicha, for her invaluable guidance, insightful feedback, and unwavering support. Her expertise and encouragement have been instrumental in shaping this research.

I am profoundly grateful to my family and friends for their continuous support and encouragement. Their understanding and patience have been a source of strength throughout this journey.

I would also like to express my sincere appreciation to my colleagues and professors at University 8 May 1945 for their support and for providing a stimulating academic environment. Special thanks to the students of 2nd year Master SIQ, whose advice and collaboration were essential to this success.

Furthermore, I would like to acknowledge the Director of Public Equipment in Guelma, the Head of Administration, and My Work Drills for their support and contributions.

Lastly, I extend my gratitude to all the participants and contributors to this study. Their cooperation and insights were critical to the successful completion of this work.

Thank you all for your support and encouragement.

And special thanks to Dr LAMKACHIR.S

Dedication

I dedicate this work

To my father, my mother and my grandmother

To My brothers,My sisters and their Kids

To my Friends

Abdellah

Abstract

This Master's thesis presents a comparative study of four popular free data mining tools: RapidMiner, Weka, KANIME, and Orange. The study evaluates their features, user-friendliness, performance, and community support. It examines data preparation, clustering and classification, and performs static and dynamic studies on various datasets. The study also assesses the integration and expansion capabilities of each tool. The results show significant disparities in performance and usefulness, with RapidMiner and Weka showing strong performance in managing large datasets and complex tasks. KANIME and Orange, on the other hand, offer intuitive interfaces and seamless connectivity with other data mining tools. This study provides valuable insights for data scientists, researchers, and practitioners in choosing the best data mining technology for their needs. By understanding each tool's unique characteristics and constraints, users can make informed choices that enhance their data analysis processes and support evidence

Keywords: Data Mining tools, RapidMiner, Weka, KANIME, Orange, Comparative Study, Machine Learning algorithms.

الملخص

تقدم مذكرة الماستر دراسة مقارنة لأربع أدوات التنقيب عن البيانات مجانية مشهورة RapidMiner, Weka, KNIME, Orange. تقيم الدراسة ميزاتها وسهولة استخدامها وأدائها ودعم المجتمع. وهو يفحص إعداد البيانات وتجميعها وتصنيفها، ويقوم بدراسات ثابتة وديناميكية على مجموعات بيانات مختلفة . تقيم الدراسة أيضا قدرات التكامل والتوسع لكل أداة تظهر النتائج تفاوتات كبيرة في الأداء و الفائدة، حيث أظهر RapidMiner and Weka أداءً قويا في إدارة مجموعات البيانات الكبيرة والمهام المعقدة. من ناحية أخرى توفر KANIME and Orange واجهات بديهية واتصالا سلتا مع أدوات معالجة البيانات الأخرى توفر هذه الدراسة رؤى قيمة لمحللي البيانات والباحثين والممارسين في اختيار أفضل تقنية التنقيب عن البيانات لتلبية احتياجاتهم. من خلال فهم الخصائص والقيود الفريدة لكل أداة، يمكن للمستخدمين اتخاذ خيارات مستنيرة تعزز عمليات تحليل البيانات الخاصة بهم .

الكلمات المفتاحية: ادوات التنقيب عن البيانات، RapidMiner, Weka, KNIME, Orange.، دراسة مقارنة، خوارزميات التعلم الآلي.

Contents

List of Figures

List of Tables

General Introduction

Over the last decade, there has been an exponential interest in data collection on different application domains, resulting in a rising demand for organizing and translating masses of facts into valuable information in various data sources. Extracting large amounts of data is crucial for gaining insights, making informed decisions, and driving innovation across various industries. The increased interest in extracting meaningful knowledge from data for the benefit of the data owner has resulted in the development of several data mining software tools. Indeed, the literature covers various tools and software with varying degrees of applicability. There are several ways in which these tools and programs differ from one another. One is in the classification of the methods they offer, which can be either statistical or based on machine learning. Another is in the data extraction phases that they incorporate, which can be either preprocessing, extraction, or validation. But for a user with a data game who wants to run a knowledge extraction procedure, this may be an inconvenience. If the preprocessing and learning methods that the user wants to implement on their data set are in different tools, and even different versions of the same tools, then he will face the problem of importing and exporting intermediate and final results and data between these tools since each of them proposes its format for the data and results. Additionally, there is a performance fluctuation problem among these many tools.

This problem prompted us to conduct a comparative study of the main free data mining software tools, which are RapidMiner, Weka, KANIME, and Orange. These tools are the most common in the professional and academic communities due to their comprehensive features, ease of use, and strong support from their respective user bases [1]. Even if commercial software sometimes offers better products, we have limited ourselves to free software due to the availability and to avoid the licensing problem. This study provides valuable insights for data scientists, researchers, and practitioners in choosing the best data mining technology for their needs.

The remainder of the present Master's thesis is structured as follows:

Chapter One: A Review on Data Mining: It consists of an introduction and definition of data mining and general basic concepts. Ends with its application areas.

Chapter two: Data Mining tools: The second chapter is an overview of the tools used in our study, which are: RapidMiner, Weka, KANIME, and Orange.

Chapter three: Comparative study: It presents our contribution. It gives results on static and dynamic analysis of data mining tools used in our study.

The general conclusion summarizes our contribution, presenting the key findings and insights.

Chapter 1

A Review On Data Mining

1.1 Introduction

Data mining is the process of obtaining valuable information from vast datasets. The process entails applying a range of methodologies from statistics, machine learning, and database systems to detect patterns, correlations, and trends within the data. Subsequently, this data can be utilized to make judgments based on empirical evidence, resolve issues within a commercial context, and reveal concealed insights. Data mining is utilized for several purposes such as consumer profiling and segmentation, market basket analysis, anomaly identification, and predictive modeling. Data mining methods and technologies are extensively utilized in diverse areas, such as banking, healthcare, retail, and telecommunications.

In this chapter, we present the essentials of the data mining domain including, the KDD process, data mining process and tasks with tools of data mining.

1.2 History of Data Mining

Data mining, sometimes referred to as knowledge discovery in data (KDD), is the systematic process of extracting patterns and other important information from a large dataset. Due to the continuous development of data warehousing technology and the exponential expansion of big data, the use of data mining methods has significantly surged over the last several decades. This has greatly aided enterprises in converting their unprocessed data into valuable insights. Nevertheless, even while technology consistently advances to manage large-scale data, leaders still have difficulties with scalability and automation[2].

Data mining has enhanced corporate decision-making by conducting artificial intelligence algorithms. The data mining methods that support these investigations may be categorized into two primary objectives: (1) Describing the target dataset or (2) Predicting results using machine learning algorithms. These strategies are used to arrange and sift through data, bringing to the forefront the most captivating information, ranging from fraud detection to user habits, bottlenecks, and even security breaches [3].

By integrating data analytics and visualization technologies, exploring the realm of data mining has become more accessible, and the extraction of relevant insights has become more expedient.

The progress in artificial intelligence continues to accelerate its implementation across several sectors. Data mining allows the dissection of a voluminous database to support opinions when traditional query languages are unworkable. Data mining is a shape of knowledge detection essential for working cases in a special sphere. There's distraction about the exact meaning of the tours "data mining" and knowledge detection in databases "KDD" [3].

In the 1990s, the term "Data Mining" was introduced, but data mining is the evolution of a sector with an extensive history.

Early techniques of identifying patterns in data include the Bayes theorem (1700s), and the evolution of regression(1800s). The generation and growing power of computer science have boosted data collection, storage, and manipulation as data sets are broad in size and complexity level. Explicit hands-on data investigation has progressively been improved with indirect, automatic data processing, and other computer science discoveries such as neural networks, clustering, genetic algorithms (1950s), decision trees(1960s), and supporting vector machines (1990s) [4].

1.3 KDD Process

The knowledge discovery in data (KDD) process is iterative at each stage, inferring that moving forward to the former conduct might be needed. Therefore, it's demanded to understand the process and the nonidentical conditions and possibilities in each stage.

The process begins with arbitrating the KDD objects and ends with the discovered knowledge. This closes the circle, and the impacts are also measured on the new data. Figure 1.1 shows the KDD process.

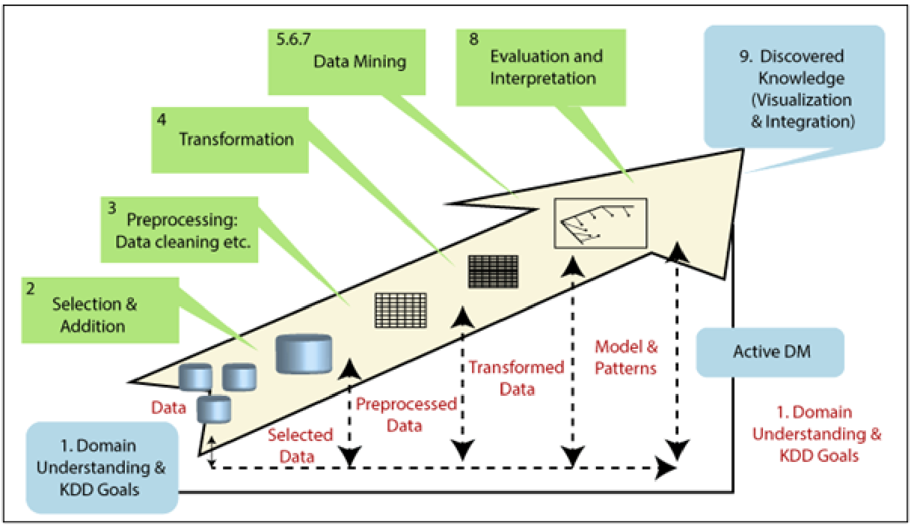

5.6.7
Data Mining
8
Evaluation and Interpretation
9. Discovered Knowledge (Visualization & Integration)
4
Transformation
3
Preprocessing: Data cleaning etc.
2
Selection & Addition
Data
Selected Data
Preprocessed Data
Transformed Data
Model & Patterns
Active DM
1. Domain Understanding & KDD Goals
1. Domain Understanding & KDD Goals

FIGURE 1.1: KDD Process [4].

1.3.1 Domain understanding

Initial preliminary step is the scene for understanding what should be done with the various decisions like transformation, algorithms, representation, etc. The individuals who are in charge of a KDD venture need to understand and characterize the objectives of the end-user and the environment in which the knowledge discovery process will occur (involves relevant prior knowledge).

1.3.2 Data selection

After establishing the objectives, it is necessary to identify the data that will be used for the process of knowledge discovery. This method entails identifying available data, acquiring relevant information, and then consolidating all the data for the purpose of knowledge discovery into a single dataset, which will be evaluated based on certain criteria. Data mining is a crucial activity as it enables the learning and discovery from the available data. This is the empirical foundation for constructing the models. If certain crucial traits are absent, then the entire study may prove fruitless in this regard, particularly when more attributes are taken into account. However, the process of organising, collecting, and managing sophisticated data repositories is costly, and there is a framework available for gaining the most comprehensive understanding of the phenomenon. This arrangement pertains to the interactive and iterative nature of the KDD process.

1.3.3 Preprocessing

During this stage, the dependability of the data is enhanced. The process includes data cleansing, such as managing missing values and eliminating noise or outliers. In this context, it may involve the application of intricate statistical techniques or the use of a Data Mining algorithm. For instance, if there is a suspicion that a particular attribute lacks dependability or has several missing data, it can become the target of the supervised algorithm in Data Mining. A predictive model will be developed for these characteristics, enabling the prediction of missing data. The extent to which one focuses on this phase of expansion depends on various circumstances. However, analysing the many components is constantly enlightening in relation to enterprise data systems.

1.3.4 Data transformation

Developing and preparing suitable data for Data Mining occurs at this stage. Here, methods include both dimensional reduction (useful for things like feature extraction and record sampling) and attribute transformation (useful for things like functional transformation and discretization of numerical characteristics). The success of the whole KDD project hinges on this stage. As an example, medical evaluations often place more emphasis on the quotient of traits than on any one of them alone. Potentially uncontrollable consequences, efforts, and temporary problems are all things to consider in business. Looking into the effects of advertisement buildup is one example.

1.3.5 Data mining

Next step is to choose a Data Mining technique, such as classification, regression, clustering, etc. This is mostly dependent on the KDD goals and the preceding stages. In data mining, there are two main goals: providing descriptions and making predictions. The former encompasses the unsupervised and visualisation components of Data Mining, whereas the latter is more often known as supervised data mining for prediction.

A large variety of data mining approaches rely on inductive learning, which involves building a model either explicitly or implicitly from a set of preparation models. The inductive method is based on the premise that the model may be used for similar circumstances in the future. The approach additionally considers the meta-learning degree of the available data set.

Finally, the Data Mining algorithm has been implemented. During this stage, it may be necessary to employ the algorithm multiple times before a satisfactory result is achieved.

1.3.6 Evaluation and interpretation

The evaluation is related to assessing and analysing the extracted patterns, rules, and their reliability to the objective defined in the initial step. For instance, incorporate a characteristic in the data transformation phase, then continue the process from that point onwards. This step is centred on evaluating the clarity and usefulness of the generated model. During this stage, the acquired knowledge is also documented for future utilisation. The final stage involves the utilisation of Data Mining, as well as the collection of feedback and the acquisition of overall discovery outcomes.

1.3.7 Discovered knowledge

The knowledge becomes efficacious in that we can modify the system and assess the consequences. The successful completion of this step determines the efficacy of the entire Knowledge Discovery in Databases (KDD) process. There are several obstacles associated with this stage, including the loss of the controlled "laboratory conditions". For instance, knowledge previously derived from a certain static representation, typically in the form of data, but now the data has become dynamic.

1.4 Data storage

Before introducing various data storage methods, it's essential to elucidate the distinctions between Data, Information, and Knowledge [5].

1.4.1 Difference between Data, Information and Knowledge

Data is any facts, text, or numbers without any context or interpretation that a computer can process [6]. According to the Oxford *"Data is distinct pieces of information, usually formatted in a special way"*. Data can be measured, collected, reported, and analyzed, whereupon it is often visualized using graphs, images, or other analysis tools. Raw data "unprocessed data" may be a collection of numbers or characters before it's been "cleaned" and corrected by researchers. It must be corrected so that we can remove outliers, instruments, or data entry errors. Data processing commonly occurs in stages, and therefore the "processed data" from one stage could also be considered the "raw data" of subsequent stages. Field data is data that's collected in an uncontrolled environment. Experimental data is the data that is generated within the observation of scientific investigations. Data can be generated by: Humans, Machines or Human-Machine.

Data is represented with the help of characters such as alphabets (A-Z, a-z), digits (0-9) or special characters (+,-,/,*,<,>,= etc.)

Information is organized or classified data, which has some meaningful values for the receiver. Information is the processed data on which decisions and actions are based [6]. For the decision to be meaningful, the processed data must qualify for the following characteristics:

- Timely: Information should be available when required.
- Accuracy: Information should be accurate.
- Completeness: Information should be complete.

Knowledge is a mix of contextual information, experiences, rules, and values. It is richer and deeper than information and more valuable because someone has thought deeply about that information and added his or her own unique experience, judgement, and wisdom. Knowledge also involves the synthesis of multiple sources of information over time. The amount of human contribution increases along the continuum, from data to information to knowledge. The more complex and ill-defined elements of knowledge are difficult, if not impossible to capture electronically.

Although knowledge has always been important to the success of an organization, it was presumed that the natural, informal flow of knowledge was sufficient to meet organizational needs. But managing knowle-dge has become far more complex, the amount of knowledge to manage far greater than ever, and the tools to manage knowledge far more powerful. Managing knowledge provides value to organizations in several ways [6].

1.4.2 Databases

Databases are purposefully created for the execution of transactions and are constructed to efficiently manage instantaneous interactions. Consequently, companies rely on OLTP [7] database systems, which are abbreviated as online transaction processing and are employed to oversee routine transactional activities that involve operational data, such as customer interactions and sales records. Relational databases are extensively utilized in diverse industries and applications, making it one of the most prevalent types of databases [7] .

Relational databases have gained popularity over several decades because of their organized format, presented in a table-like structure, and their effective handling of structured data using SQL (Structured Query Language). Relational database management systems such as Oracle, MySQL, PostgreSQL, and Microsoft SQL Server are some examples. Over the course of time, various sophisticated databases have been developed, including object-relational databases, object-oriented databases, multimedia databases, logical databases, and distributed databases [8].

Nevertheless, when confronted with escalating data complexity and volume, conventional relational databases have difficulties in effectively managing exceedingly big datasets. In response to this, a solution known as NoSQL (Not Only SQL) databases has been created. NoSQL databases are specifically designed to handle unstructured and semi-structured data. They achieve great scalability by distributing data across numerous servers or clusters and allow flexibility by not enforcing a particular format. NoSQL databases encompass several types such as key-value stores, document stores, column-family stores, and graph databases. NoSQL databases such as MongoDB, Cassandra, Redis, and Neo4j are some notables examples [7].

1.4.3 Data warehouse

Data warehouse is an enabled relational database system designed to support very large amounts of historical data from various sources and are optimized for complex queries and data analysis. When relational databases are used for OLTP systems, data warehouses are based on OLAP [7] for online analytical processing, which is designed to support complex data analysis and reporting.

Data in the data warehouse is organized around specific subjects and aggregated from multiple sources [5]. Historical data is stored to support trend analysis and comparison over time. Thus, once data is loaded into the warehouse, it is not frequently updated, ensuring data integrity. Examples of data warehouse solutions include IBM Netezza, Amazon Redshift, Google Big Query, and Snowflake.

Data warehouses are often used for business intelligence and data mining purposes. Business intelligence (BI) includes technologies, and processes used by enterprises for analyze data and make decisions [5]. BI aims to turn raw data into actionable insights, helping organizations make informed choices to enhance their operations and strategic direction. Data warehouses allow analysts to access and analyze data from multiple databases and sources in a centralized and consistent manner.

A data mart is a subset of a data warehouse that focuses on specific business areas of the corporation [5]. It is intended to simplify retrieving relevant and consolidated data for analytical and reporting needs.

1.5 Data mining process

Cross-Industry Standard Process for Data Mining (CRISP-DM) is a widely used methodology for conducting data mining or analytics projects. It provides a structured approach to guide professionals through the various stages of a data mining project, from understanding business objectives to deploying the model into production [9]. Figure 1.2 depicts the data mining process.

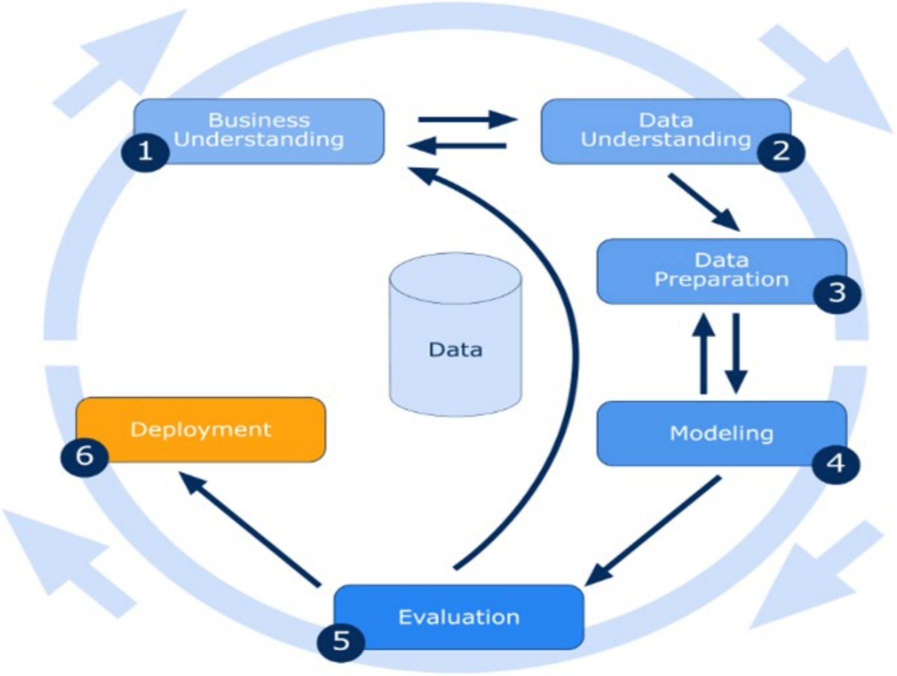


FIGURE 1.2: Cross Industry Standard Process for Data Mining [10]

.

1.5.1 Business understanding

During the business knowledge stage, it is important to determine the problem that we want to solve utilizing data mining (e.g., how to create a more targeted marketing campaign). The business issue description, filled by stakeholders other than data scientists, will generate the project's main inquiries. Further study is required to fully comprehend the business environment. Defining project objectives and success criteria is necessary prior to collecting and evaluating correct data.

1.5.2 Data understanding

After the business problem has been identified, it is necessary to ascertain the kind of data required and locate pertinent sources. Data scientists gather information in this stage from a variety of sources, including customer databases and transaction records. Not every data piece, though, could be pertinent to the project. For example, a business could only be interested in credit card purchases. Ensuring that only pertinent material is provided is the aim here. The data mining team ought to have chosen the subset of data required to solve the issue by the time the data understanding phase was over.

1.5.3 Data preparation

Typically consumes about 90 per cent of the time of the project. It is the longest stage and requires many steps to prepare the data for additional processing and analysis. This might entail data cleansing, or removing outliers, duplicates, and missing data from the dataset. To be ready for the next step, data from many sources may be combined, arranged, or modified in various ways. This step determines which factors are most important and prepares the final data set.

1.5.4 Modeling

To improve results, the Modeling phase repeatedly refines the model development procedures by using various data mining approaches. By refining the models with each iteration, stakeholders are given more authority to make educated choices using data-driven insights. To construct information systems and manage data efficiently, data modelling is essential. To aid with the development and deployment of databases and related systems, it generates graphical depictions of data structures, relationships, and regulations. Finding important entities, describing their properties and connections, and selecting suitable modeling approaches are all part of this process.

1.5.5 Evaluation

At this critical stage, we evaluate the created machine learning models against a set of clear criteria. When it comes to choosing the best models and tweaking hyper-parameters, these indicators provide us with the quantifiable data we need. A common important metrics that are used for assessment include:

- **Loss function:** This statistic measures the difference between the actual labels and the projected values. Model performance is improved with lower loss levels.
- **Accuracy:** The percentage of occurrences that are successfully predicted is measured by accuracy, a key parameter for classification tasks.
- **Precision and recall:** Precision is how many out of all positive predictions were really right, whereas recall measures how many out of all positive cases the model properly detected.
- **F1-score:** is a balanced measure that takes into account both recall and precision; it gives a single number to evaluate the overall performance of the model.

The comparison of these assessment outcomes to the project-initiated objectives to ensure alignment with the original business goals and check whether models provide the expected results and make a good contribution to the company strategy in this way.

1.5.6 Deployment

Insights gained via data mining may enhance corporate results and decision-making. To be successful, these insights must be provided in a way stakeholders can understand and access. It's important to provide insights so stakeholders may use them to make choices.

The complexity of data mining implementation depends on the company's needs. It might be a report summarizing the results or a more complex and repeatable data mining process throughout the enterprise. Deployment, maintenance, and monitoring procedures are needed to keep insights valuable and relevant, regardless of complexity. When deploying, consider both initial installation and ongoing support and maintenance. This requires implementing monitoring and maintenance plans to resolve difficulties and prolong the installed solution.

The manager of the project should write a report on the data mining project's results. This report covers project results, problems, and improvements. Lessons and insights from the study may guide future data mining activities in the organization.

In summary, stakeholders must be presented with data mining insights, business deployment requirements must be considered, and project evaluations must be done to enhance data mining methods. The structured CRISP-DM [11] paradigm is used across sectors to address data mining tasks[12].

1.6 Types of data mining

Data mining can also be used to extract insights from unstructured data, such as text, webs, multimedia, graphs and images, using techniques such as natural language processing and computer vision [5].

1.6.1 Text Mining

Text mining involves the study of statistics, data mining, machine learning, and Natural language processing (NLP) of the text data type [13]. For tasks like sentiment analysis, document summarizing, text classification, and text clustering, text mining is a lifesaver. Using NLP and machine learning algorithms can extract valuable information from the text. To do this, methods like statistics are used to uncover the latent trends and patterns. Preprocessing the text using stemming and lemmatization to transform textual data into data vectors is necessary for text mining.

1.6.2 Multimedia Mining

Image data, video data, audio data, and the complex relationships between them are all part of multimedia data objects. Finding interesting patterns in all these different datasets is the focus of multimedia data mining. Image processing, classification, data mining for audio and video, and the complex world of pattern recognition are all part of this endeavour's high-level digital data processing. Because of its far-reaching effects on so many different kinds of social media, multimedia data mining is quickly becoming one of the most interesting areas of study. By using this field, we may extract trends and insights from social media sites like Twitter and Facebook, revealing fascinating patterns in the digital fabric of our linked world [14].

1.6.3 Web mining

Finds vital information and patterns on the Internet. Data from various websites, including web pages and multimedia data like photos on web pages, may be analyzed using online content mining. Web mining is carried out to get knowledge on the following: web page content, unique users, unique hypertext linkages, rating and relevancy of web pages, summaries of web page content, user search trends, and the amount of time users spend on a certain website. Web mining can also help you choose the finest search engine and learn how it uses its algorithm. Finding the finest search engine for users and improving search efficiency are two of its main functions. There are three main categories of web mining: Web Usage Mining, Web Content Mining, and Web Structure Mining [15].

Web usage Mining Focuses on analyzing user interactions and behaviors while they navigate websites, for example log files.

Web Content Mining Focuses on extracting information and knowledge, from the textual content present on web pages. Example, the content of web page.

Web Structure Mining Focuses on analyzing the relationships and link structures among web pages. Example, use Page Rank algorithm to assign a score to web pages based on the number and quality of links.

1.6.4 Graph Mining

Discovering the hidden connections in complex graphs or networks is the main goal of graph mining. It has an impact on many aspects of network behavior. Some examples are recommendation networks, social networks, and citation networks. By treating nodes and edges as separate things and the relationships between them, graph mining can process structured data. The interconnections and dynamics of these networks may be better grasped with the aid of this picture. From inspecting individual nodes and edges to discovering emerging patterns and motifs, graph mining uses a variety of tools and techniques to examine the network. Each node contains vital information and becomes a focal point of inquiry. However, edges also serve as windows into the network, showing how connections and interactions form its structure [16].

1.7 Data mining branches

Each of the subsequent data mining strategies serves several distinct business challenges and offers unique insights into each of them. However, comprehending the specific sort of company challenge you must address will also aid in determining the most suitable strategy to use, which will ultimately produce the most optimal outcomes. The forms of Data Mining may be categorised into two fundamental components, which are as follows [17]:

- Predictive Data Mining Analysis
- Descriptive Data Mining Analysis
- Prescriptive Data Mining Analysis

1.7.1 Descriptive Analytics

Descriptive data Analyses finds patterns and similarities in data rather easily. To find relevant clusters in the given data, descriptive data mining may be used as well. Data preparation for reports and analysis is the primary goal of this mining technique. It tells you things about the data, including the average and count. Without any prior knowledge, it reveals what is occurring within the data. It displays the data commonalities [18].

1.7.2 Predictive Analytics

Predictive Analytics focus on future forecasts rather than on past actions. It uses the target-prediction skills acquired via supervised learning. This sub-field of data mining encompasses methods including regression, classification, and time-series analysis. Developers benefit from this as it clarifies the features that are not immediately apparent.

One example is comparing the results of past quarters business analyses to those of the next. Predictive analysis often makes use of existing data to infer or forecast features [19].

1.7.3 Prescriptive Analytics

Descriptive analytics informs us about past events, whereas predictive analytics forecasts future events. Prescriptive analytics, on the other hand, advises on the optimal course of action to be taken. This methodology represents the ultimate and most advanced step in the analytical process of corporate affairs.

Prescriptive data mining prompts firms to take action and assists executives, managers, and operational personnel in making optimal decisions by utilising the available data [20].

1.8 Data Mining tasks

1.8.1 Classification

The classification sorts things in a collection according to certain attributes. The system is trained to predict the category of objects from an unknown collection of items using a training set that contains items with known attributes. In order to label a set of objects or make a prediction about a class, it employs techniques such as KNN (K Nearest Neighbor), decision trees, etc. [21].

1.8.2 Regression

The regression problem involves creating a model that predicts continuous numerical values or a vector of values, rather than discrete classes, based on input attributes. This prediction can be made using classical statistical methods, advanced statistical methods, or symbolic methods commonly used in classification tasks. An example of a regression model is: "there is a linear relationship between the unit price and quantity purchased when buyer A buys energy"[22].

1.8.3 Clustering

Clustering is an unsupervised learning task that involves identifying clusters of comparable items in data that have common attributes. It may be used in data mining to assess similarities among data, create a collection of representative prototypes, examine correlations between variables, or automatically depict a dataset with a limited number of areas while maintaining the topological qualities of the original input space. It discovers clusters of interconnected data entries that serve as a foundation for delving into further connections.

Clustering approach helps in creating population segmentation models, specifically demographic-based consumer segmentation. Further analysis using typical analytical and data mining methods may identify the properties of these segments in relation to a certain intended result.

For instance, analyzing the purchasing patterns of various population groups might help identify the segments that should be focused on for a new sales campaign [23] .

1.8.4 Association

An association rule is a set of literals expressed as X Y, where X and Y are sets of things, derived from a collection of transactions where each transaction is a set of literals. This rule suggests that if a database transaction includes X, it is likely to also include Y.

Association rule generators are a potent data mining method used to explore an entire dataset for rules that expose the type and frequency of linkages or associations among data elements. The generated associations may filter data for human examination and perhaps establish a prediction model using observed behavior [24].

1.8.5 Summarization

The summarization means to generalize or abstract the facts. The result is a condensed set that mostly contains aggregate data but still provides some insight into the data. There are a variety of summarization formats, including numerical format (ex. means, max, standard deviations, etc.), graphical format (ex. histograms and scatter plots), and rule-based ("if-then") [25].

1.8.6 Pattern recognition

Pattern recognition is the ability of computers to identify patterns in data, and then use those patterns to make decisions or predictions. Pattern recognition can be miscellaneous information: text, images, emotions, sentiments, sounds, symbols, numbers, etc. It is based on statistics and machine learning techniques and it is very significant in a lot of domains such as multimedia, sonar, speech recognition, vision and agriculture [26].

1.9 Data mining techniques

Many methods from different fields, including statistics, machine learning, pattern recognition, information retrieval, database and data warehousing systems, visualization, algorithms, high-performance computing, and other application areas, have been integrated into data mining. One of the main reasons for the success of data mining and its wide range of applications is the multidisciplinary character of data mining research and development.

1.9.1 Hypothesis testing

Method entails formulating a hypothesis, crafting a series of queries to test that hypothesis, and then analysing the data returned by the database to discover correlations between a few variables and establish statistical significance when applied to a population. A subset of the queries is included in the reports, which comprise chosen analytical findings provided in textual, tabular, and graphical formats. A wide variety of data sources, such as databases, data warehouses, spreadsheets, etc, are accessible for querying and reporting [27].

1.9.2 Statistical methods

Since the amount of data produced is increasing exponentially and knowledge gained from understanding data enables quick and informed decisions that save time and give a competitive advantage, statistical methods have become increasingly important in identifying patterns and trends in otherwise unstructured and complex large sets of data. This is the reason why statistical techniques for data mining have made significant strides in the last several years [28].

Descriptive analysis: Statistical analysis in quantitative research begins with describing response characteristics, such as the average of a single variable or the relationship between two variables, once data collection is complete. After that, it is important to know how to use inferential statistics to determine whether your data supports or contradicts of hypothesis and if it can be applied to a wider population. It uses numerical methods such as [29]:

Minimun Ordering a data set $x_1 \leq x_2 \leq x_3 \leq \ldots \leq x_n$ from lowest to highest value, the minimum is the smallest value x1.

$$\text{Min} = x_1 = \min(x_i) \quad \text{for} \quad i = 1 \text{ to } n$$

Maximun Ordering a data set $x_1 \leq x_2 \leq x_3 \leq \ldots \leq x_n$ from lowest to highest value, the maximum is the biggest value xn.

$$\text{Max} = x_n = \max(x_i) \quad \text{for} \quad i = 1 \text{ to } n$$

Mean Indicates where the data of a distribution is centralized. It is measured by dividing the sum of all values and the number of instances [30].

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- x_i: represents each value in the dataset
- n : is the total number of values in the dataset.

Median When a dataset is arranged from lowest to highest, the median is a central tendency measure in descriptive statistics that represents the midway value. The middle value, or median, is used when the number of observations in the collection is odd. The median is the midpoint of the two middle values when the number of observations in the dataset is even. The median may be calculated in this way[30]:

- Arrange the data in ascending order.
- If the number of observations is odd:

The median is the middle value of the dataset:

$$\tilde{x} = x_{\frac{n+1}{2}}$$

- If the number of observations is even: The median is the average of the two middle values:

.

$$\text{Median} = \frac{\text{Value of } \left(\frac{n}{2}\right) \text{ th observation} + \text{Value of } \left(\frac{n}{2}+1\right) \text{ th observation}}{2}$$

Median is resilient in the face of extreme numbers and outliers. In cases when the distribution is biased or if there are outliers in the sample, it provides a more accurate picture of the central tendency.

Mode: is the value that appears most frequently in a data set. The formula for finding the mode depends on the type of data:

- For discrete data (data that can only take specific values): The mode is the value with the highest frequency.

- For continuous data (data that can take any value within a range): The mode is often identified by constructing a histogram or a frequency distribution and finding the peak(s) with the highest frequency.

Mode may be found using software for large datasets or computed manually for smaller ones. The mode is a calculation-free alternative to the mean and median; it's just the most common value or values in the information set.

Variance: is the measure of the dispersion of the data regarding the mean value of the data. It informs us how the data is scattered in the supplied data value. The formula for calculating the variance depends on whether the user is dealing with a population or a sample.

Variance (σ^2) $\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}$

where:

- x_i represents each value in the dataset,
- $\bar{x}$ is the mean,
- N the total number of observations,

The key difference between the population and sample variance formulas is the denominator. The population variance is divided by the total number of values (N), while the sample variance is divided by one less than the total number of values (N-1). This adjustment in the sample variance formula (using N-1 instead of N) is known as Bessel's correction and is used to provide an unbiased estimate of the population variance from a sample.

Standard deviation: Standard deviation (σ^2 for population and s^2 for sample) is the square root of the variance. It provides a measure of how much individual data points differ from the mean of the dataset [31]:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N}}, s = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}$$

Quartiles: Dataset is divided into four equal sections by its quartiles. If you want to know what the first, second, and third quartiles are, use the symbols Q1, Q2, and Q3. Here is how quartiles are typically calculated: Arrange the data in ascending order. Median Q1 is the median of the lower half of the dataset, which is 25%, The median Q2 is the middle value of the dataset, which is 50%. Q3 is the median of the upper half of the dataset, which is 75%, while Q4 "100%" also known as the maximum, is the highest value in the dataset.

Covariance: is a statistical measure that describes the strength and direction of a relationship between two variables. It quantifies the extent to which changes in one variable are associated with changes in another variable.

$\mathrm{Cov}(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n}$

where: $\bar{x}$ and $\bar{y}$ are the means of X and Y, respectively.

- If the covariance is positive, it indicates that as one variable increases, the other variable also tends to increase.
- If the covariance is negative, it indicates that as one variable increases, the other variable tends to decrease.
- If the covariance is close to zero, it suggests that there is little to no linear relationship between the variables.

Regression: is a statistical technique used to model and analyze the relationship between a dependent variable (often denoted as Y) and one or more independent variables (often denoted as X). It helps in understanding how the value of the dependent variable changes when one or more independent variables change [32]. There are various types of regression analysis, but two common types are:

Simple linear regression: models the relationship between a single independent variable (X) and a dependent variable (Y). The relationship is assumed to be linear, and it's represented by the equation of a straight line:

$Y = \beta_0 + \beta_1 X + \epsilon$

Multiple Linear Regression: Multiple linear regression models the relationship between two or more independent variables $(X_1, X_2, \ldots, X_n)$. and a dependent variable (Y). The relationship is assumed to be linear, and it's represented by the equation of a plane or hyperplane in higher dimensions:

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$ where:

- β_0:is the intercept.
- $\beta_1, \beta_2, ..., \beta_p$: the coefficients (slopes) corresponding to each independent variable.
- ϵ: represents the error term.

1.9.3 Inferential statistics

Inferential statistics seeks to generalise its findings from a study of a smaller subset of the population to the larger whole in the hopes of revealing some trait or pattern shared by all members of the population. People living in a City, for instance, may be asked to rate their mayor. To get a feel for the general public opinion of the mayor [33].

Sampling: An equal likelihood of selection is desirable for homogeneous populations. Stratified sampling, which involves dividing a population into subgroups and selecting samples from each group in proportion to their size, may be more successful for heterogeneous populations.

Estimation: Goal of estimation is to estimate a value for an unknown population characteristic (parameter) such as means, proportions, or variances based on the sample statistics. Both point estimators and interval estimators can be used for this task.

Confidence intervals: Assists in determining the parameters of a population. 95% confidence interval suggests that if a test is repeated 100 times with fresh samples under identical circumstances, the estimate is likely to fall inside the specified range in 95 out of 100 cases. Moreover, a confidence interval helps determine the crucial value in hypothesis testing. In addition to the above tests, additional tests used in inferential statistics include the ANOVA test, Wilcoxon signed-rank test, Mann-Whitney U test, Kruskal-Wallis H test, etc.

Hypothesis Testing: using to evaluate assumptions and derive inferences about the population based on the given sample data. The process includes establishing a null hypothesis and an alternative hypothesis, then executing a statistical test for significance. A conclusion is made based on the test statistic's value, the critical value, and the confidence intervals. A hypothesis test may be conducted as left-tailed, right-tailed, or two-tailed.

1.9.4 Machine Learning techniques

Branch of AI known as machine learning involves training algorithms on data sets to build self-learning models that can categories data and make predictions without any human input whatsoever. Many modern businesses make use of machine learning for tasks such as translating text across languages, making stock market predictions, and personalizing product recommendations to customers based on their previous purchases.

Due to the widespread use of machine learning for AI applications in today's world, the phrases "machine learning" and "artificial intelligence" are frequently used interchangeably in popular language. Yet, there is a substantial difference between the two words. Machine learning, in contrast to artificial intelligence (AI), is the process of training computers to mimic human intelligence via the analysis of existing data and algorithms[34].

There are three principal types of Machine Learning: supervised learning, unsupervised learning and reinforcement learning. A semi-supervised learning can be defined according to the two first types of machine learning.

Supervised learning

Supervised learning entails learning a mapping between a set of input variables X and an output variable Y and applying this mapping to predict the outputs of unseen data. Supervised learning is the most important methodology in machine learning[35]. By leveraging labelled data, supervised learning algorithms are trained to recognise patterns and make accurate predictions, making them a critical tool in applications ranging from image and speech recognition to medical diagnosis and financial forecasting.

Unsupervised learning: The absence of labels in the data is what defines unsupervised learning. Improving one's data literacy requires first understanding how to decipher the data underlying patterns. Clustering is a process that groups data points that are similar together without using labels. One unsupervised learning approach that may be used to minimize the amount of features in a dataset while keeping important information is dimensional reduction. This method is shown by Principal Component Analysis (PCA) [36]. The semi-supervised learning is achieved by training with both labelled and unlabeled data. This method may be useful when getting labelled data is a hassle or costs a lot of money [37].

Reinforcement Learning: Learning how to behave in a given setting to maximise reward is at the heart of this field. The data used in RL comes from machine learning algorithms that learn by making mistakes. Neither supervised nor unsupervised machine learning uses data as input.

Self-Supervised Learning: is a type of unsupervised learning that does not require manual labelling. Unsupervised learning detects data patterns like clustering, community finding, or anomaly detection. Self-supervised learning focuses on recovering missing pieces within a supervised framework[38].

Transfer learning: Basic concept of transfer learning involves applying information gained from activities with abundant labelled data to situations with limited labelled data. The prerequisites for transfer learning are specific to the task and the model being used [39].

Multi-task learning: Involves training a model simultaneously for various tasks. Deep learning models are commonly used due to their flexible adaptability. Different tasks make use of the first levels of the network, which are then followed by layers and outputs that are specialized for those tasks [39].

Deep learning: Typically, most Deep Neural Networks (DNNs) consist of a greater number of layers. DNNs are trained on extensive data to discover and categories occurrences, recognize patterns and correlations, assess possibilities, and make predictions and judgements. A single-layer neural network may provide approximate predictions and judgements, but the presence of many layers in a deep neural network enhances and fine-tunes the results for increased accuracy [40].

Data Visualization: is a crucial stage in dataset analysis. When executed precisely, it can:

- assists the user in developing a profound comprehension of the dynamics inside our dataset.
- Accelerate the machine learning component of the analysis.
- Simplify our dataset study for better comprehension by others.

One of the most famous, powerful, and well-known libraries of Python, "Matplotlib," can be use to create an animated GIF of a PCA variance plot.

1.10 Data Mining areas

Data mining has revolutionized business strategy design, enabling understanding of the present to anticipate future outcomes in various industries.

1.10.1 Marketing

Marketing has been radically altered by data mining. One benefit of data mining in marketing is that it allows companies to get real-time suggestions based on customer purchasing history. Businesses can boost their sales with these suggestions. Have you ever been shopping on Amazon and had more items suggested to you after adding an item to your cart? Those suggestions, if any, were generated by data mining algorithms. If you decide to buy more from Amazon, their sales will increase, and Businesses and marketers can access client data stored in AI-powered databases through data mining. Marketers now have more information than ever before on customer behaviour thanks to data mining in marketing. This leads to more precise predictions and improved sales. The process of market segmentation also frequently makes use of data mining.

1.10.2 Agriculture

The Farmers in the agriculture sector face a lot of issues and difficulties due to the improper understanding and implementation of the activities to enhance their growth and productivity. A large amount of data is available for analysis and scrutiny, however, those related to the agriculture sector are in a small quantity. Hence segregation and processing of the same from the sources has to be done with proper methodology. Places having multiple grain growth and different soil structure make it complex to have a perfect estimation of the crop yield both in quantity and quality. Creating a close link between customer expectation and the producing capabilities of the agriculture sector can be a win-win situation at both ends, this can be achieved by capturing data segment-wise and in a structured manner [41].

1.10.3 Energy

Power and energy systems (PES) are changing fast, taking advantage of the sophisticated information and communication infrastructure that connects the electricity “smart” grids. However, the amount of data coming from PES seems to be increasing drastically, with more and more data generated and stored daily. Thus, extracting valuable information so that humans can understand, interpret, and analyze becomes a nontrivial task that needs to be addressed to take advantage of the available data fully. Data mining, a process to find relationships between multiple variables and even discovery intrinsic structures and interconnection in large data sets, seems to be the perfect tool to overcome this issue, encompassing a set of algorithms that can be used in multiple applications in the energy field.This book chapter introduces data mining in PES, covering some of the most used and influential algorithms in the field, and providing an overview of diverse applications, such as profiling and forecasting in PES[42].

1.10.4 Healthcare

Data mining enables more accurate diagnostics. Having all of the patient's information, such as medical records, physical examinations, and treatment patterns, allows more effective treatments to be prescribed. It also enables more effective, efficient and cost-effective management of health resources by identifying risks, predicting illnesses in certain segments of the population or forecasting the length of hospital admission. Detecting fraud and irregularities, and strengthening ties with patients with an enhanced knowledge of their needs are also advantages of using data mining in healthcare.

1.11 Conclusion

Data mining finds predictive information in vast datasets. New and sophisticated technology enables firms to priorities crucial data in their warehouses. Using data mining techniques may help make informed judgements by predicting future trends and behaviours. Data mining is crucial for this reason. In the next chapter, we will delve into the specific tools and methodologies selected for our comparative study, providing a detailed analysis of their features, strengths, and applications.

Chapter 2

Data mining tools

2.1 Introduction

Data mining techniques are categorised into descriptive, inferential, predictive, and prescriptive analytics. There is a growing demand for tools that can directly analyse data and draw conclusions due to the increasing requirement for data analysis.

tools include creating a report to summarise conclusions, offer improved visualisations, and deliver precise outcomes with minimal effort. Various tools for data mining include RapidMiner, Weka, KNIME, Orange, Open Refine, Solver, Julia, etc. Our contribution tends to make a comparative study between four open-source data mining tools: RapidMiner, Weka, KNIME, and Orange. We will determine the most efficient tool based on specific factors. Our study aims to conduct a comparative analysis of four open-source data mining tools: RapidMiner, Weka, KNIME, and Orange. Our objective is to identify the most efficient tool based on specific criteria.

In this chapter, we will examine the graphical interface and components of each tool in detail. We will explore the layout, usability, and features of the graphical user interface (GUI), as well as the various components such as menus, toolbars, panels, and workflow editors.

2.2 RapidMiner

RapidMiner may not have the name recognition of AWS or Google, but it is a comprehensive data science platform. It aids organizations in exploring, blending and cleansing data, designing and refining predictive models through machine learning and managing deployments [43]. For businesses looking for a robust, expansive ML toolset, RapidMiner bears exploring.

RapidMiner uses a unified interface to manage various tasks though a graphical drag-and-drop approach. It offers pre-defined machine learning libraries but also incorporates numerous third-party libraries. This includes hundreds of components encompassing machine learning, text analytics, predictive modeling, automation and process control.

RapidMiner produces a fast classification and regression analysis system for both supervised and unsupervised learning. The solution also supports split and cross-validation methods that improve the accuracy of predictive models. Both Gartner and Forrester rank RapidMiner as a "Leader." The vendor also earned a Gartner Customer's Choice 2018 award.

2.2.1 Installation

Follow these instructions to download RapidMiner Studio: To download the application, go to the RapidMiner website, then resources click on download to access to download page. Search for the version that matches your operating system. Click on Free Download.

Running the installation

1-Double-click on the downloaded file.

2-If prompted, allow the program to make changes to your computer. The RapidMiner Studio Setup Wizard appears. Click Next to continue.

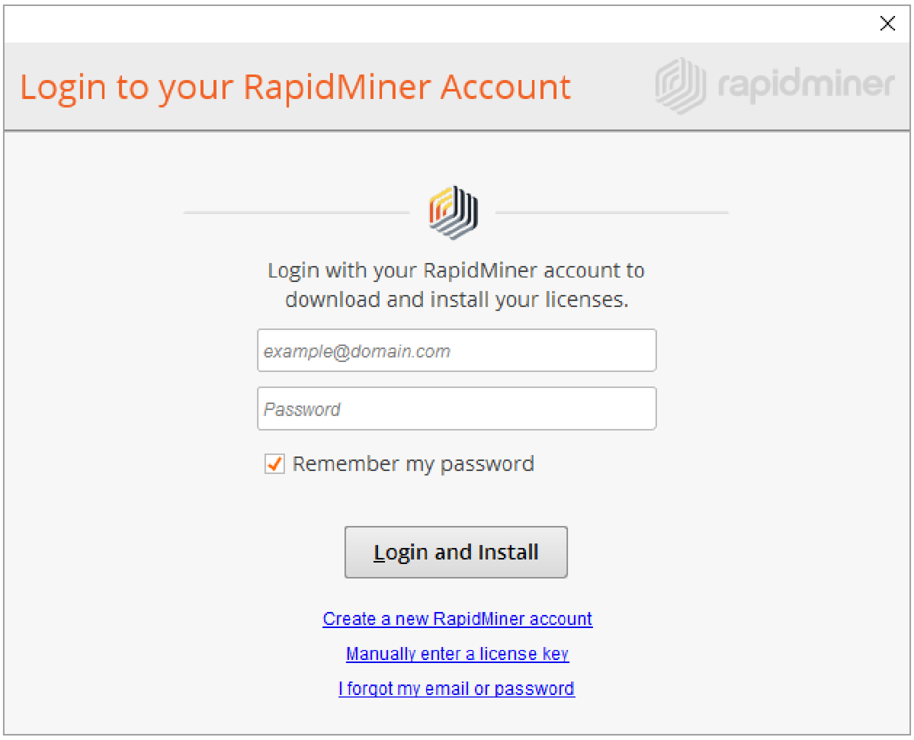
3-Read the terms of the license agreement and click I Agree to continue.

4-Select a destination folder (or leave the default).*Please ensure that the folder path does not contain + or characters*. By clicking Install, the wizard extracts and installs RapidMiner Studio. When the installation completes, click Next and then Finish to close the wizard and start RapidMiner Studio.

5-Read the terms of the license agreement and click I Accept to continue [44].

2.2.2 How to use

to Getting started with RapidMiner Studio Once you launch RapidMiner Studio, and you have accepted the EULA, a Welcome screen appears, prompting you to login with your RapidMiner Account.



Login to your RapidMiner Account

Login with your RapidMiner account to download and install your licenses.

example@domain.com

Password

☑ Remember my password

Login and Install

Create a new RapidMiner account

Manually enter a license key

I forgot my email or password

FIGURE 2.1: Rapidminer studio login interface

1-**Login and Install** user can enter your existing RapidMiner account credentials in the username/password fields, and download your license immediately.

2-**Create a new RapidMiner account** If user does not yet have a RapidMiner account, a new browser window will open in which he can create it.

3-**Manually enter a license key** If use does not have access to the Internet, he can install a license key manually here.

RapidMiner Studio login If the user had already created a RapidMiner account, he can log in straight away:

1-Enter his email address and password to login with his RapidMiner.com account and then click Login and Install.

Note: Once installed, he can also add or update licenses using the Settings > Manage Licenses menu within the RapidMiner Studio application.

If he can not access to his RapidMiner.com account (for example, if RapidMiner is blocked by a firewall), his license can not be automatically loaded.

Create RapidMiner account: Creating an account requires access to the Internet[45]:

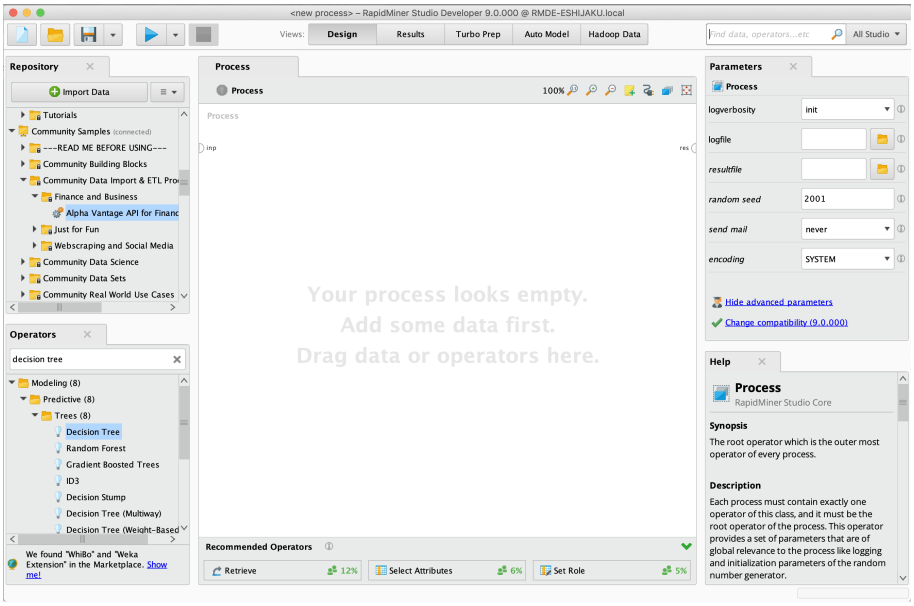
Complete the steps to create an account by following the instructions on screen.

Once he is done, he can return to RapidMiner Studio and enter his credentials to proceed.

The Start Page: If the user needs additional guidance, or he needs to accelerate his data preparation and model building, he tries Turbo Prep, RapidMiner's tool for interactive data preparation, and Auto Model, RapidMiner's solution for automated machine learning. If he needs to see more examples, chooses from one of the templates in the Samples Repository. If he needs to do it himself, creates a new (blank) process from scratch in the Design View.

RapidMiner: includes numerous panels (see Figure 2.2):

- Data, processes, and results are stored in the ***Repository.***
- The essential elements of every workflow are called***Operators.***
- Operators are connected via ***ports***. The output of the first is passed as input to the second.
- A connected set of Operators that help you to transform and analyze your data is called a ***process***.
- The behavior of an Operator can be modified by changing its ***parameters***.
- The behavior of an Operator can be understood by reading the ***Hepl***.


<new process> – RapidMiner Studio Developer 9.0.000 @ RMDE-ESHIJAKU.local
Views:
Design
Results
Turbo Prep
Auto Model
Hadoop Data
Find data, operators...etc
All Studio
Repository
Import Data
Tutorials
Community Samples (connected)
---READ ME BEFORE USING---
Community Building Blocks
Community Data Import & ETL Pro
Finance and Business
Alpha Vantage API for Financ
Just for Fun
Webscraping and Social Media
Community Data Science
Community Data Sets
Community Real World Use Cases
Operators
decision tree
Modeling (8)
Predictive (8)
Trees (8)
Decision Tree
Random Forest
Gradient Boosted Trees
ID3
Decision Stump
Decision Tree (Multiway)
Decision Tree (Weight-Based
We found "WhiBo" and "Weka Extension" in the Marketplace. Show me!
Process
Process
100%
inp
res
Your process looks empty.
Add some data first.
Drag data or operators here.
Recommended Operators
Retrieve 12%
Select Attributes 6%
Set Role 5%
Parameters
Help

Process

logverbosity	init
logfile	
resultfile	
random seed	2001
send mail	never
encoding	SYSTEM

Hide advanced parameters

Change compatibility (9.0.000)

Process
RapidMiner Studio Core

Synopsis

The root operator which is the outer most operator of every process.

Description

Each process must contain exactly one operator of this class, and it must be the root operator of the process. This operator provides a set of parameters that are of global relevance to the process like logging and initialization parameters of the random number generator.

FIGURE 2.2: RapidMiner panels

Process: A connected set of Operators that help the user to transform and analyze your data. Also known as: ***flow, program, pipeline, diagram***. When user have connected all his ***Operators*** and set their ***parameters*** , press the Run ***Run*** arrow button at the top of the user interface, and the results will be displayed in the ***Results View***. There is more than one way to run your process:

- ***Locally***.
- ***In the background***.
- On RapidMiner ***AI Hub***.
- On RapidMiner AI Hub, as a ***scheduled process***.

As user processes grow in size, he will need some way to manage their complexity.

- User can***hide the complexity***, by moving groups of Operators into a single *Subprocess* Operator.
- user can run ***a process from within another process***, via *the Execute Process* Operator.

- To save your process to a ***Repository***, *select File > Save Process* from the main menu.
- User can easily share a process by first exporting it to an **XML** file:
- To export the process, ***select File > Export Process***. The export dialog allows user to save the file as *.rmp* or *.xml*; in reality, both these file formats are identical (***XML***).
- To import the process, select ***File > Import Process***.

A simple process, where the data from an Excel file is (1) read, (2) stored in the Repository, and (3) displayed in the Results View.

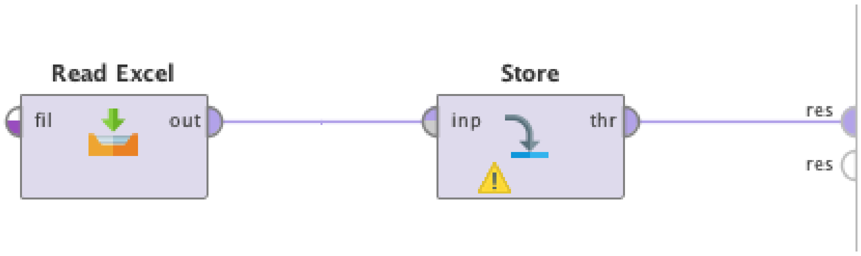


FIGURE 2.3: simple process

Repository: Central data storage entity. It holds connections, data, processes and results, either locally or remotely. Also known as: ***folder, workspace, project***.

When working with RapidMiner Studio, user need a place to save his work. The Repository can be used to store: Connections, Data, Processes, Results, etc.

Project As of RapidMiner Studio 9.7, a Project supports both version control and arbitrary file types. It behaves in the same way as a Repository, but with the addition of version control.

As discussed in the Projects documentation for RapidMiner Studio and RapidMiner AI Hub, a Project always has both a local component and a server component, and the two are regularly synchronized.

2.2.3 Operators

The elements of a Process, each Operator takes input and creates output, depending on the choice of parameters. Also known as: ***function, formula, node***.

To use RapidMiner Studio effectively, user has to learn about its Operators. RapidMiner Studio includes hundreds of Operators, and therefore a large part of the task is learning how to find what he needs. As so often with search, there are two major strategies: hierarchical search and keyword search. The RapidMiner Community is also a source of support.

To verify that the Operator he has found has the functionality he expects, read the Help.

Once he's found the Operator you want, there are at least 3 ways of getting it into the Process Panel.

- Drag and drop the Operator
- Double-click the Operator
- Right-click the Operator, and choose Insert Operator from the context menu.

Hierarchical search: The hierarchy of folders in the Operators Panel reflects a typical data science workflow:

- Data Access
- Blending
- Cleansing
- Modeling
- Scoring
- Validation
- Utility

- Extensions

By opening these folders and their subfolders, user will get some insight into what's available.

This same hierarchy can be examined on the docs website, which includes the Help for each Operator.

Keyword search: The alternative is keyword search. Although the Operators Panel includes a search field, the recommended procedure is to use the global search, in the upper right corner of the user interface. The global search finds not just Operators, but data and processes from the Repository, extensions from the Marketplace, and even actions you can take from the menu!

Community search (Wisdom of Crowds): If user has started building a process, and he has looking for hints, the "Wisdom of Crowds" can be helpful. The "Wisdom of Crowds" is an opt-in recommender system, based on the usage pattern of other RapidMiner users. It predicts which Operators you might need, based on the Operators that are already included in his process. To activate it, click on the button that says Activate Wisdom of Crowds. he can activate it or deactivate it at any time via the menu item ***Settings > Preferences > Recommender > Enable operator recommendations***.

Parameters: *Options for configuring the behavior of an Operator* . The content of the Parameters Panel is context-dependent. Select any Operator that is displayed in the Process Panel, and the Parameters Panel displays the options for configuring that Operator. Because RapidMiner Studio includes many Operators, each with its own unique functionality, the range of parameters is also quite diverse. By default, RapidMiner Studio will show you only the more commonly used parameters. To see all of the available parameters, click Show advanced parameters.

To understand the parameters, user need to learn more about the Operator; reading the Help for that Operator is probably a good place to start. Alternatively, hover the information icon ⓘ next to the parameter of interest, and a help text is displayed.

All of the Operator help texts provided within RapidMiner Studio are also available online.

Reconfiguring the Design View:

To restore the Design View to the default panel setup, select ***View > Restore Default View***.

To optimize your screen real estate, you might consider reorganizing the panels. Notice first that you can right-click the tab connected with any panel, and select one of the following[46] :

- **Detach** : The panel is detached from RapidMiner Studio.
- **Maximize** : The panel fills the entire space allotted to panels.
- **Close**: The panel is removed from the user interface

2.3 Weka

Weka is a collection of machine-learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rule mining, and visualisation. Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like this, and the bird sounds like this.

The "Waikato" Environment for Knowledge Analysis, abbreviated as Weka, may be accessed as Java source code on the website www.cs.waikato.ac.nz/ml/weka. This is a comprehensive and robust implementation that contains code examples and functional applications of machine learning techniques. It provides clear and concise versions of basic methods, intended to help comprehend the underlying mechanics. The tool also offers a workbench with fully functional, cutting-edge implementations of several popular learning algorithms for practical data mining or research purposes. The software includes a Java class library framework that assists applications using embedded machine learning and allows for the creation of new learning algorithms[47]. It includes tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also applicable for producing new machine learning schemes.

One method of using Weka is to use a learning approach to a dataset and analyze its output to learn more about the record. The second is to need learned models to make predictions on new instances. A third is to use multiple learners and compare their performance to select one for prediction. In the interactive Weka interface, it can choose the learning method it is required from a menu. Several methods have tunable parameters, which can create through a property sheet or object editor. A common computation structure is used to compute the performance of all classifiers [48].

2.3.1 Installation

Folllow the below steps to install Weka on Windows:

Step 1: Visit this website using any web browser [49]. Choose the version that matches your operating system. Click on Free Download.

Step 2: It will redirect to a new webpage, click on Start Download. Downloading of the executable file will start shortly. It is a big 118 MB file that will take some minutes.

Step 3: Now check for the executable file in downloads in your system and run it.This executable will install Weka in your Program Menu. Launching via the Program Menu or shortcuts will automatically use the included JVM to run Weka

Step 4: It will prompt confirmation to make changes to your system. Click on Yes.
Step 5: Setup screen will appear, click on Next.

Step 6: The next screen will be of License Agreement, click on I Agree.

Step 7: Next screen is of choosing components, all components are already marked so don't change anything just click on the Install button.

Step 8: The next screen will be of installing location so choose the drive which will have sufficient memory space for installation. It needed a memory space of 301 MB.

Step 9: Next screen will be of choosing the Start menu folder so don't do anything just click on Install Button.

Step 10: After this installation process will start and will hardly take a minute to complete the installation.

Step 11: Click on the Next button after the installation process is complete.

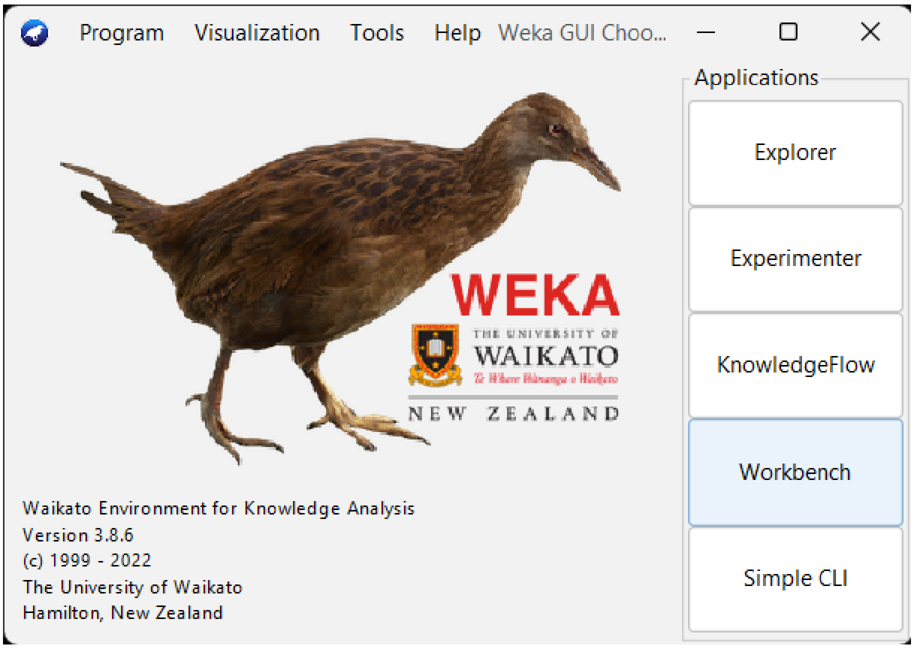
Step 12: Click on Finish to finish the installation process.

Step 13: Weka is successfully installed on the system and an icon is created on the desktop.

Step 14: Run the software and see the interface.

2.3.2 How to use

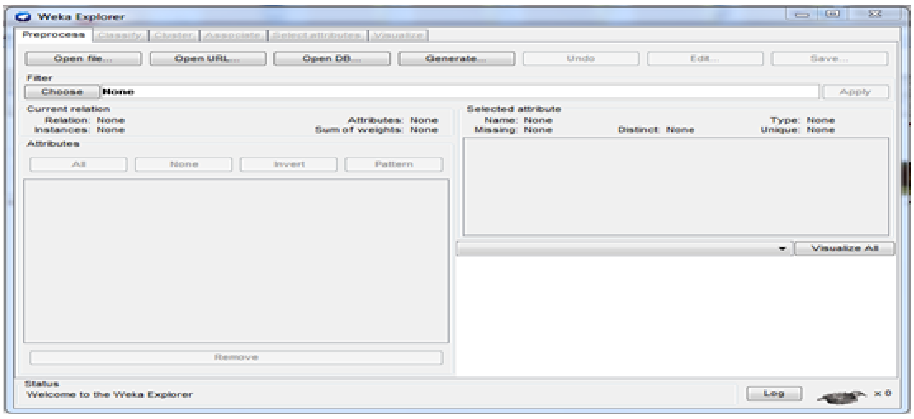
To get start with Weka double click on Weka icon or one click on all programs–Weka. Weka has its own file format called ARFF (Attribute Relation File Format), but can also process data from relational databases (SQL DB), binary, CSV (Open File) files, or upload files to the Web (Open URL).



Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

FIGURE 2.4: Weka interface

It has several tabs that give access to the main components of the workspace,the next figure show the Explorer principal interface of Weka.



Filter: None

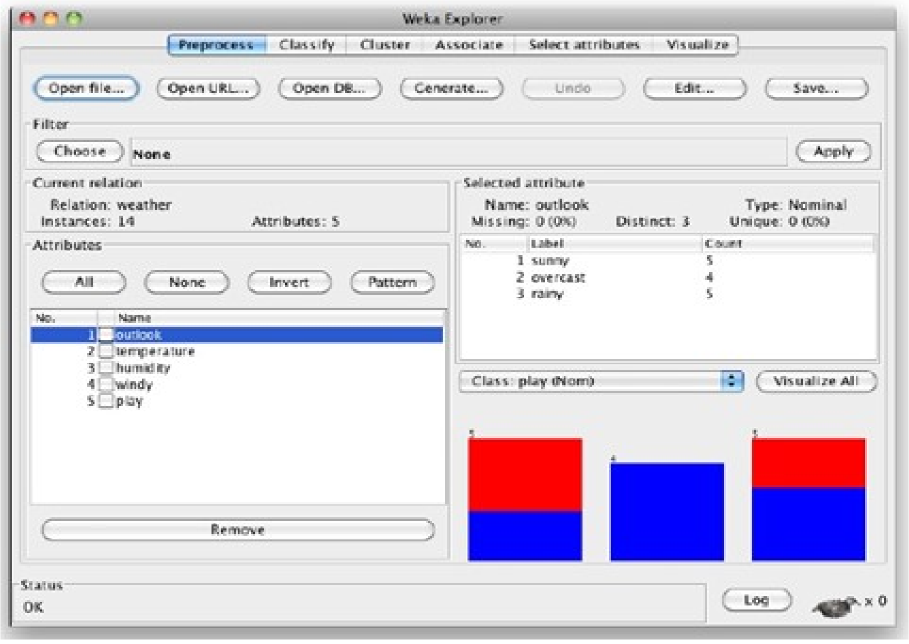
Relation: None
Instances: None
Attributes: None
Sum of weights: None

Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Status
Welcome to the Weka Explorer

FIGURE 2.5: Interface explorer.

The Preprocess tab (Preprocess, Classify, Cluster, Associate, Select Attributes and Visualize) Under these tabs, there are several pre-implemented machine learning algorithms are available in the Preprocess tab, also known as the "preprocessor." We can load files in the ARFF format, which is specific to Weka, or data from databases or a CSV file, and then apply a filtering algorithm to preprocess the data. With the help of these filters, you can do things like convert real-numerical properties to discrete ones and remove instances and attributes based on certain criteria. The Preprocess tab's interface is shown in Figure 2.6.



Filter: None

Relation: weather
Instances: 14 Attributes: 5

No.		Name
1	□	outlook
2	□	temperature
3	□	humidity
4	□	windy
5	□	play

Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

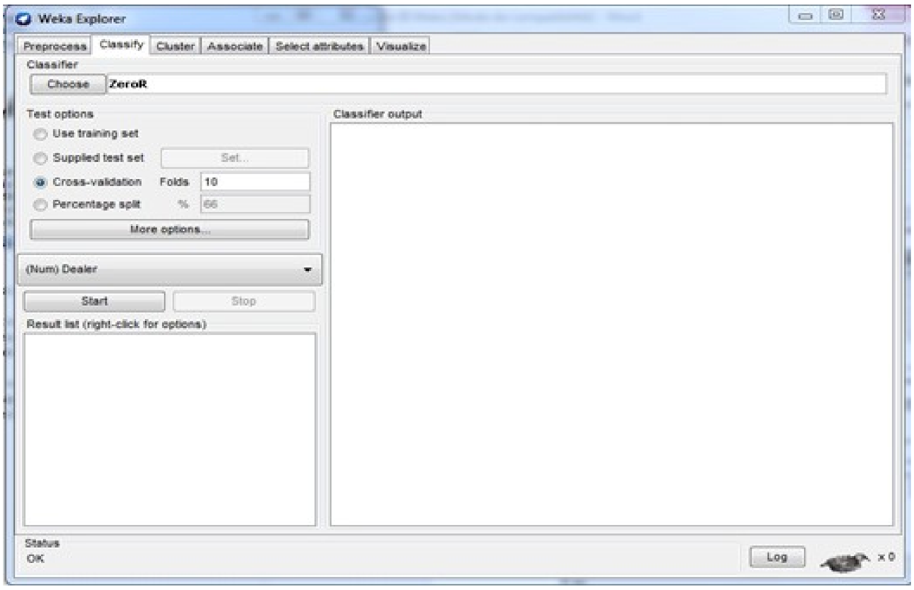
No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

Class: play (Nom)

Status: OK

FIGURE 2.6: Preprocess tab.

Classify tab After the data has been loaded, the Classify tab lets the user apply regression and classification algorithms to the dataset. They can then estimate the predictive model's accuracy and even see the model's mistakes or successes visually, like in a Decision Tree. the next figure shows the Classify tab user interface.



Classifier: ZoroR

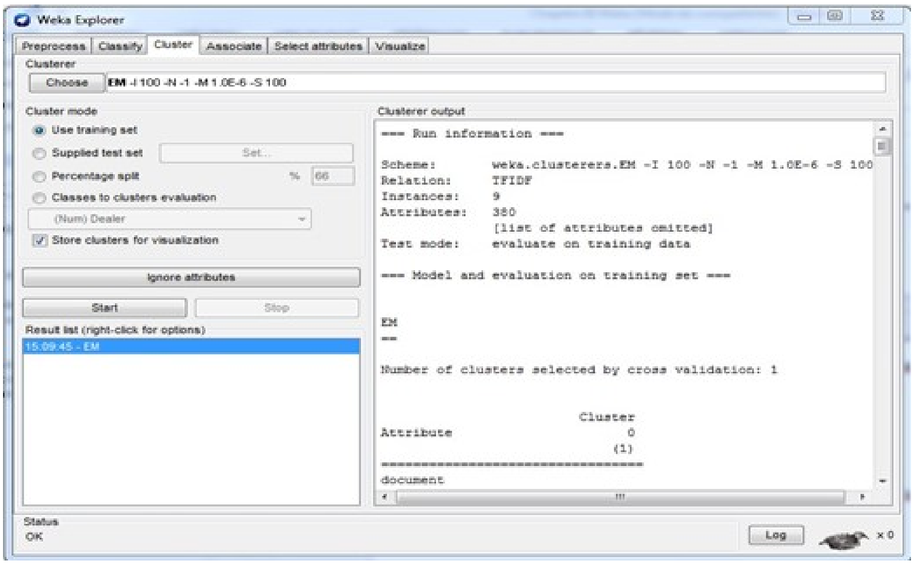
Test options:
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Num) Dealer

Status
OK

FIGURE 2.7: Classify tab

Cluster tab The Tab Cluster provides access to the clustering methods (segmentation techniques) of Weka, such as the K-means algorithm. The interface of the tab Cluster is shown in figure 2.8.



Clusterer: EM -I 100 -N -1 -M 1.0E-6 -S 100

Cluster mode:
● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Num) Dealer
☑ Store clusters for visualization

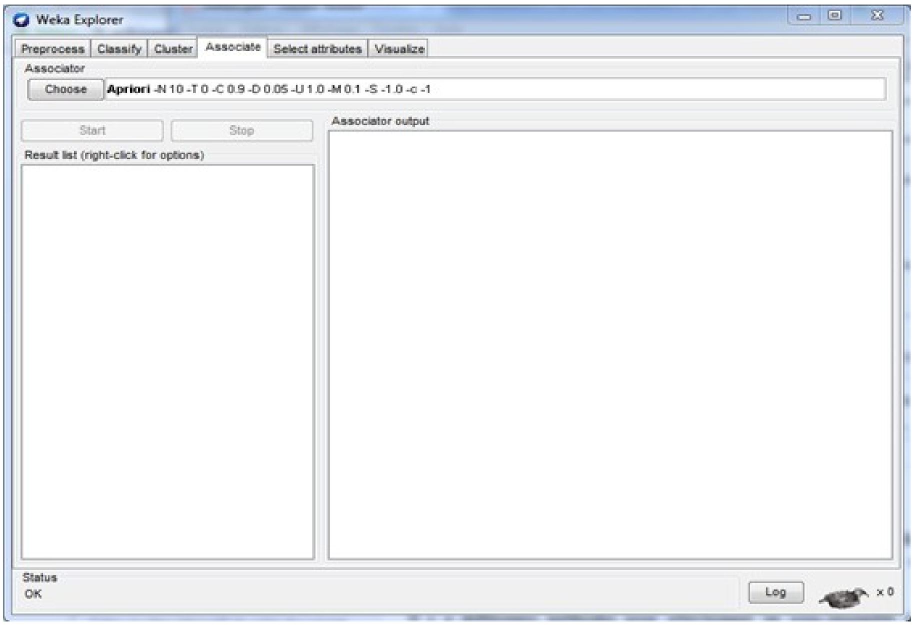
15:09:45 - EM

```
=== Run information ===

Scheme:       weka.clusterers.EM -I 100 -N -1 -M 1.0E-6 -S 100
Relation:     TFIDF
Instances:    9
Attributes:   380
              [list of attributes omitted]
Test mode:    evaluate on training data

=== Model and evaluation on training set ===


EM
==

Number of clusters selected by cross validation: 1


                           Cluster
Attribute                        0
                               (1)
=================================
document
```

Status
OK

FIGURE 2.8: Cluster tab

Associate tab: contains the association rules

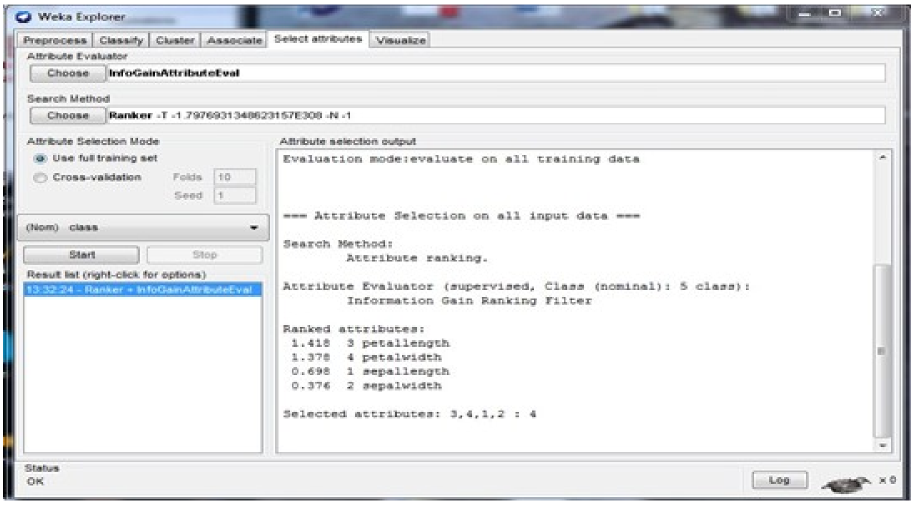


Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

FIGURE 2.9: Associate tab

Select attributes tab

The "Select attributes" page offers techniques for determining the most influential features in a dataset and enables the selection of attributes to be utilized for categorization. Various techniques exist for choosing a subset of characteristics to use in the categoriz-ation process. This technique is very beneficial in situations when the data is excessively noisy, including several qualities that do not contribute to the classification process. Performing a thorough cleaning (selection) would be really advantageous in this particular situation. This option also enhances the pace of treatments. To do this, one must choose the Info Gain Attribute Eval method in the "Attribute Evaluator" and the Ranker method in the "Search Method." The Info Gain Attribute Eval method evaluates the attributes based on their information gain, while the Ranker method arranges the attributes according to their value. By selecting the Ranker option, you have the ability to define the criteria for selection. This may be done by establishing a threshold or by specifying the number of characteristics to retain. The Ranker feature allows for the identification of the most predictive features within a dataset and facilitates the selection of attributes to be used for classification.The interface of the Select Attribute tab is shown in Figure 2.10.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options): 13:32:24 - Ranker + InfoGainAttributeEval

Attribute selection output

```
Evaluation mode:evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 class):
        Information Gain Ranking Filter

Ranked attributes:
 1.418  3 petallength
 1.378  4 petalwidth
 0.698  1 sepallength
 0.376  2 sepalwidth

Selected attributes: 3,4,1,2 : 4
```

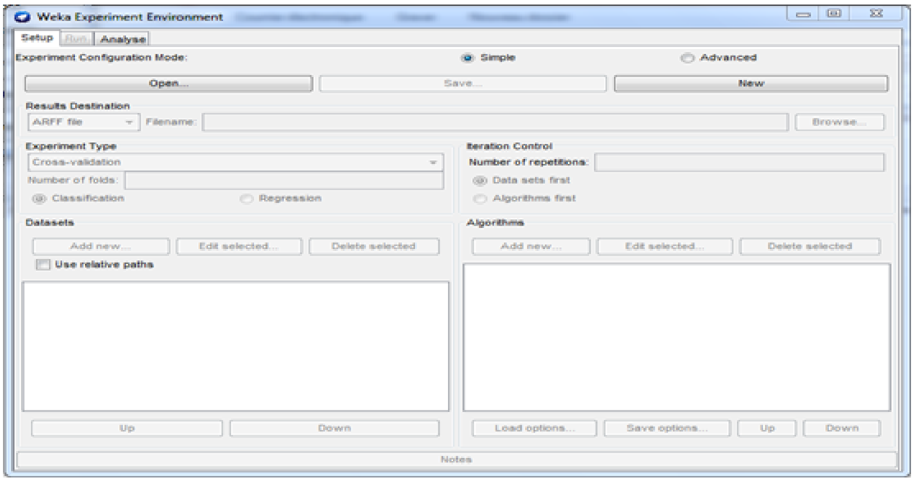
Status: OK

FIGURE 2.10: Select attributes tab

Visualize tab

The last tab, "Visualize," displays a matrix of scatter plots, where individual scatter plots may be selected and enlarged, and further analyzed using various selection operators. The Visualize window has a set of 25 graphics, each representing a view of the whole set of examples based on two potential dimensions, with the color of the points indicating their class. On the graph, each point represents an example: you may get the description of that example by clicking on it. The color of a point corresponds to its class.

Experimenter The experimenter facilitates the methodical evaluation (taxonomy) of the prediction capabilities of Weka's machine learning algorithms on a set of datasets. Figure 3.5 displays the Experimenter interface of Weka.



Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination
ARFF file Filename:

Experiment Type
Cross-validation
Number of folds:
● Classification ○ Regression

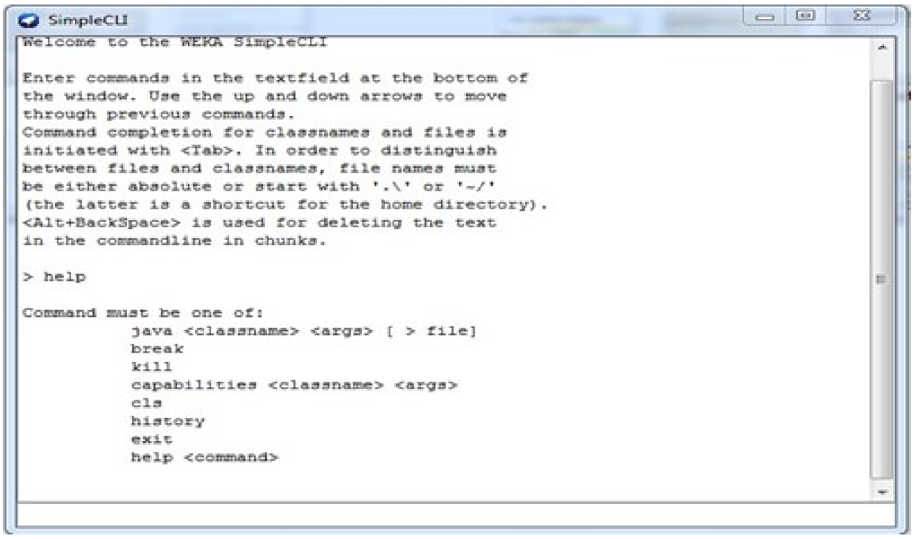
Iteration Control
Number of repetitions:
● Data sets first
○ Algorithms first

Datasets
□ Use relative paths

Algorithms

FIGURE 2.11: Experimenter Interface

CLI The SimpleCLI interface, a command line interpreter, is strongly recommended for ongoing use because of its additional features and much decreased memory requirements.The picture below depicts the Command Line Interface (CLI) of Weka.



```
Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with '.\' or '~/'
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.

> help

Command must be one of:
          java <classname> <args> [ > file]
          break
          kill
          capabilities <classname> <args>
          cls
          history
          exit
          help <command>
```

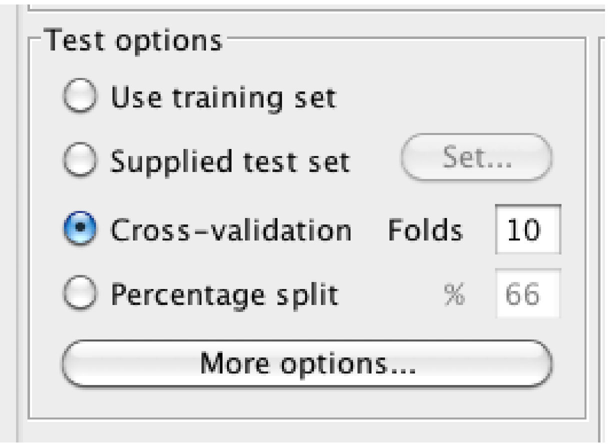
FIGURE 2.12: simple CLI

2.3.3 Operators

Data preprocessing Data preprocessing in Weka involves the use of filters, which enable the modification of datasets by performing actions such as deleting or adding properties, resampling, or deleting samples. Weka provides a comprehensive range of filters (located under the Preprocess tab), which include attribute filters that are often used for data preprocessing prior to learning. Weka implements a very large number of classifiers (based on rules, trees, Bayesian networks, etc.). Among these classifiers are:

- **ZeroR** : majority class rule;
- **J48** : decision tree;
- **NaiveBayes** : Bayes naïf ;
- **IBk** : KNN

Model Validation There are different methods of model validation in Weka, which can be found on the Test options section of the Classify tab as illustrated below



Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

FIGURE 2.13: Test options window

.

Use training set Utilize the training set, which aims to train a tree by utilizing all of the training instances. This process results in the creation of a tree and the classification outcome on the same dataset

Supplied test set (with parameter set – choice of dataset for validation): consists in evaluating the model on another dataset (a priori different from the one used to build the model).

Cross-validation (with «folds» as the parameter): to achieve the cross-validation effect, which is to partition the data into parts. This is done in cases where the training and test subsets are not already separated in the collection.

Cross-validation involves splitting the data into n groups, with the models being built on n-1 groups and tested on the nth group. Then the test group is changed, and the process is repeated until all combinations are processed. The final validation is then the average of the validations.

Percentage split The purpose of using the parameter "percentage" is to use a portion of the data to construct the model and reserve the remaining portion for validation

2.4 KANIME

KANIME is a powerful free open source data mining tool which enables data scientists to create independent applications and services through a drag and drop interface. It can serve well as a business intelligence resource, which can be used for business intelligence and data analytics.The software is available as a free download on their website.

For the purposes of direct marketing, KNIME will allow converting multiple data sources, spreadsheets, flat files, databases, and more into a standard format. This data can be normalized, analyzed, and configured to generate visual representations. In other words, it can shape data into information. This data aggregation provides the possibility to create easy to understand visualizations.

KNIME can be used as a key component of their marketing technology stack by direct marketers to gain better understanding of the large amounts of data involved with a direct marketing operation.

Many business intelligence features are built in. There are numerous data visualization tools which can be used for creating larger applications, and with some configuration, it can create an extremely powerful dashboard for analyzing direct marketing data.

Getting started with KNIME takes some configuration; it is not an out-of- the-box solution. There are numerous templates that can be configured for a multitude of purposes, however there are no direct-marketing specific ones. That said, the functionality is certainly possible to configure to meet Business Intelligence needs for use in direct marketing operations. The modular nature of KNIME makes it possible to create brand-new workflows which can be well-adapted to a BI dashboard. There are many useful features and modules that do not need to be built from scratch; in many cases it merely requires configuring the data itself to use pre-existing structures.

Once configured, this will enable marketers to create various different types of reports, and can theoretically help gain a much better understanding of users and target markets [50].

2.4.1 Installation

Go to the download page on the KNIME.com website to start installing KNIME Analytics Platform.

The download page shows three tabs which can be opened individually:

Register for Help and Updates: here user can optionally provide some personal information and sign up to our mailing list to receive the latest KNIME news

Download KNIME: this is where user can download the software

Getting Started: this tab gives user information and links about what he can do after he had installed KNIME Analytics Platform

Now open the Download KNIME tab and click the installation option that fits user operating system.

Read and accept the privacy policy and terms and conditions. Then click Download.

Once downloaded, proceed with installing KNIME Analytics Platform: for Windows: Run the downloaded installer.

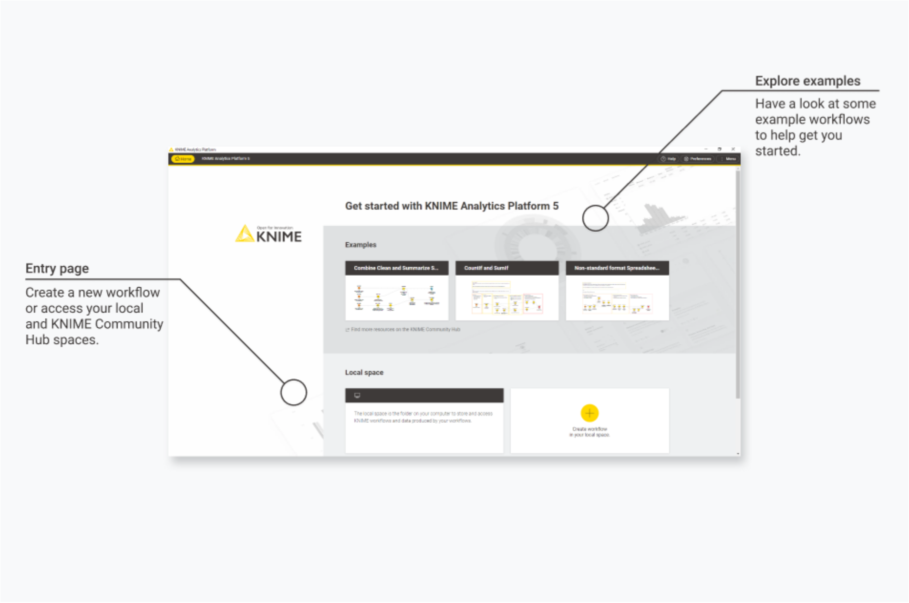
The following operating systems versions are supported:

Windows 10, 11

Windows Server - 2016, 2019, 2022.

2.4.2 How to use

Entry page: The entry page is displayed by clicking the *Home* tab



Explore examples

Have a look at some example workflows to help get you started.

Entry page

Create a new workflow or access your local and KNIME Community Hub spaces.

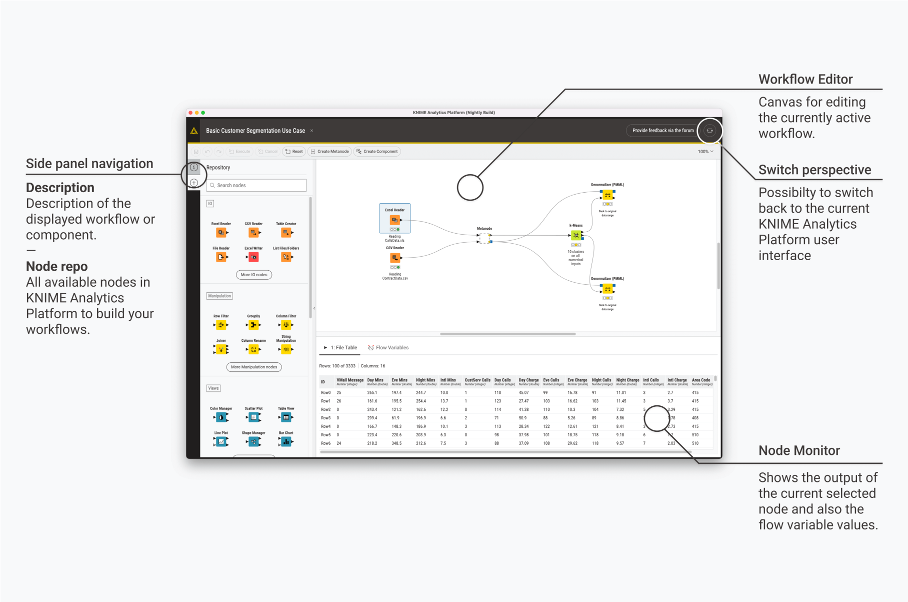
The local space is the folder on your computer to store and access KNIME workflows and data produced by your workflows.

KNIME Analytics Platform
Home
KNIME Analytics Platform 5
Help
Preferences
Menu
Open for Innovation
KNIME
Get started with KNIME Analytics Platform 5
Examples
Combine Clean and Summarize S...
CountIf and SumIf
Non-standard format Spreadshee...
Find more resources on the KNIME Community Hub
Local space
Create workflow
in your local space.

FIGURE 2.14: kNIME modern UI entry page

User interface

The active workflow of the KNIME Analytics Platform will be displayed after switching from an opened workflow. If user had open multiple workflows before he switches the perspective, only the active workflow of the current KNIME Analytics Platform will be displayed in the KNIME Modern UI. If user now closes this workflow in the KNIME Modern UI, the next opened workflow will be displayed. This happens until all workflows are closed — the entry page to create or open workflows will then be displayed.



Side panel navigation

Description
Description of the displayed workflow or component.

—

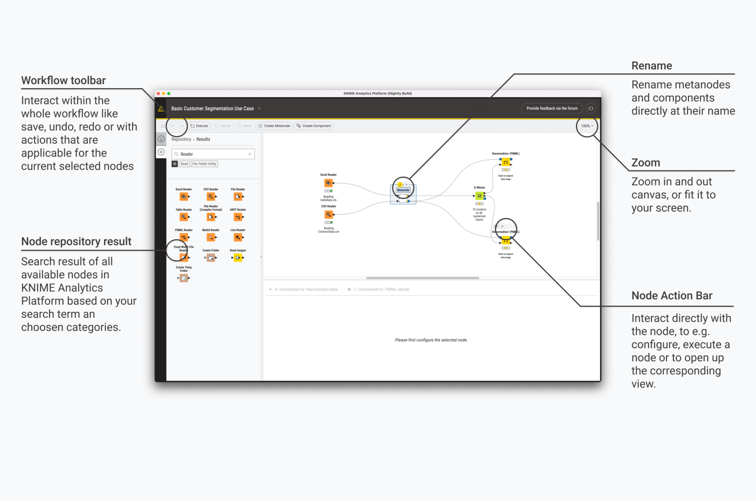
Node repo
All available nodes in KNIME Analytics Platform to build your workflows.

Workflow Editor
Canvas for editing the currently active workflow.

Switch perspective
Possibilty to switch back to the current KNIME Analytics Platform user interface

Node Monitor
Shows the output of the current selected node and also the flow variable values.

FIGURE 2.15: kNIME modern UI general layout



Workflow toolbar

Interact within the whole workflow like save, undo, redo or with actions that are applicable for the current selected nodes

Node repository result

Search result of all available nodes in KNIME Analytics Platform based on your search term an choosen categories.

Rename

Rename metanodes and components directly at their name

Zoom

Zoom in and out canvas, or fit it to your screen.

Node Action Bar

Interact directly with the node, to e.g. configure, execute a node or to open up the corresponding view.

FIGURE 2.16: Important user interface elements

2.4.3 Operators

In KNIME, operators are referred to as "nodes." These nodes are the fundamental building blocks used to construct data analysis workflows in KNIME Analytics Platform. Nodes perform specific tasks, such as data manipulation, transformation, analysis, visualization, and integration. Users can create workflows by connecting nodes together in a visual environment without the need for coding.

Here are some common types of nodes in KNIME:

Reader Nodes: Nodes for reading data from various sources such as files (e.g., CSV, Excel, XML), databases (e.g., SQL), Hadoop Distributed File System (HDFS), and web services.

Data Manipulation Nodes: Nodes for cleaning, filtering, transforming, and aggregating data. These nodes include options for handling missing values, sorting, joining, splitting, and reshaping data tables.

Data Analysis Nodes: Nodes for performing statistical analysis, exploratory data analysis (EDA), and descriptive analytics. These nodes provide functionalities for calculating summary statistics, generating histograms, scatter plots, and other visualisations, and detecting patterns in the data.

Modelling Nodes: Nodes for building predictive models using machine learning algorithms such as decision trees, random forests, support vector machines, logistic regression, and neural networks. These nodes allow users to train models, evaluate their performance using cross-validation or holdout validation, and apply them to new data for prediction.

Ensemble Nodes: Nodes for creating ensemble models by combining multiple base models using techniques such as bagging, boosting, or stacking.

Data Mining Nodes: Nodes for performing data mining tasks such as association rule mining, clustering analysis, and anomaly detection.

Integration Nodes: Nodes for integrating KNIME workflows with external systems, databases, or web services. These nodes enable data import/export, database querying, and web scraping functionalities.

Visualization Nodes: Nodes for visualizing data and analysis results in various formats such as charts, graphs, scatter plots, heatmaps, and interactive dashboards

2.5 Orange

Orange is a comprehensive C++ library that serves as a core object and routine library. It encompasses a wide range of both standard and non-standard techniques for machine learning and data mining. It is a tool that is freely available for use, which allows for the display of data, the extraction of useful information from data, and the application of machine learning techniques. Orange is a programmable platform designed for rapid development and experimentation with cutting-edge algorithms and testing methodologies. It is a collection of Python modules that are part of the core library. It utilizes some features that do not need immediate execution time and is implemented using the Python programming language [51].

The system has a range of functions, including the aesthetically pleasing display of decision trees, bagging and boosting techniques, attribute subset selection, and several more. Orange is a collection of graphical widgets that incorporates methods from the core library and orange modules to provide a satisfactory user interface. The widget facilitates digital communication and may be assembled into an application using a visual programming tool known as an orange canvas.

Collectively, these attributes render an orange a distinctive algorithm that is built on components and utilizes data mining and machine learning. Orange is designed for proficient users and data mining analysts who want to develop and evaluate their own algorithms while maximizing code reuse. It is also suitable for newcomers to the subject who can use Python scripts for data analysis.

The purpose of Orange is to provide a framework for conducting experiments, developing predictive models, and creating recommendation systems based on data analysis. It is largely used in the fields of biology, genetic research, healthcare, and education. It is used in education to enhance the teaching of data mining and machine learning to students studying biology, and biomedical.

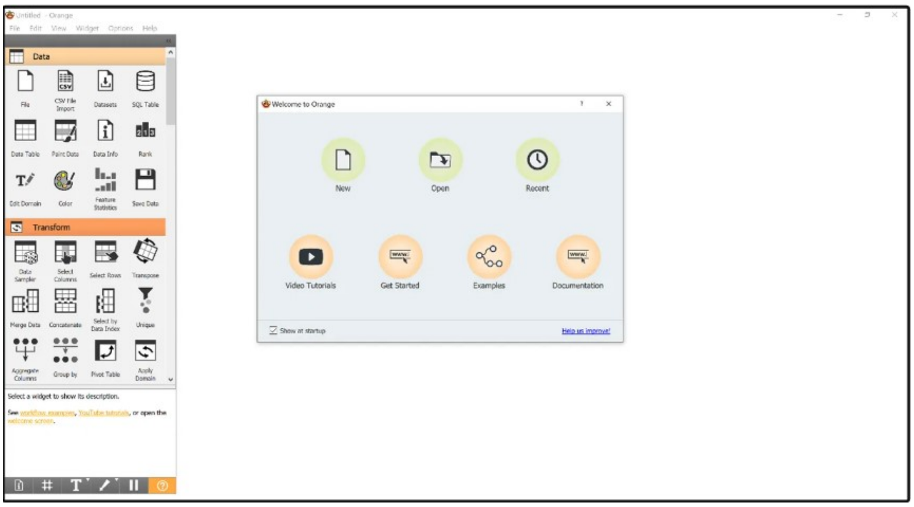
2.5.1 Installation

1- download Orange 3.36.2 from official site [52].

2- Double-click on the downloaded file, accept the license agreement then choose users, then choose components and choose install location after choose start menu folder and click next.

2.5.2 How to use

One Orange tool is installed Opening the tool interface will look like this:



Select a widget to show its description.

See workflow examples, YouTube tutorials, or open the welcome screen.

FIGURE 2.17: Orange first run interface

The popup enables user to initiate a new workflow from scratch or open an existing one. Upon clicking the 'New' icon, a blank canvas is revealed.

To start populating this empty canvas, utilize the widgets, which serve as the computational units in Orange Data Mining.

2.5.3 Operators

Access them through the explorer bar on the left, conveniently organized into subgroups based on their functions:

1-Data: for loading and storing data

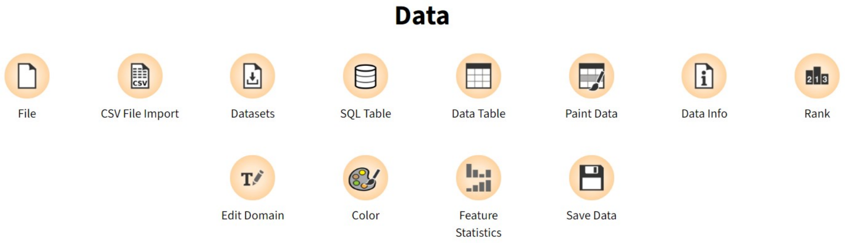


FIGURE 2.18: Data Widget

.

2-Transform: useful for data preparation and transformation

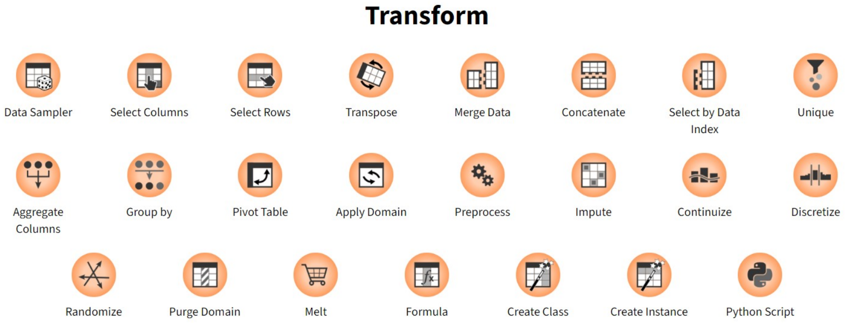


FIGURE 2.19: Transform-Widget

.

Visualize: to create visualizations from data

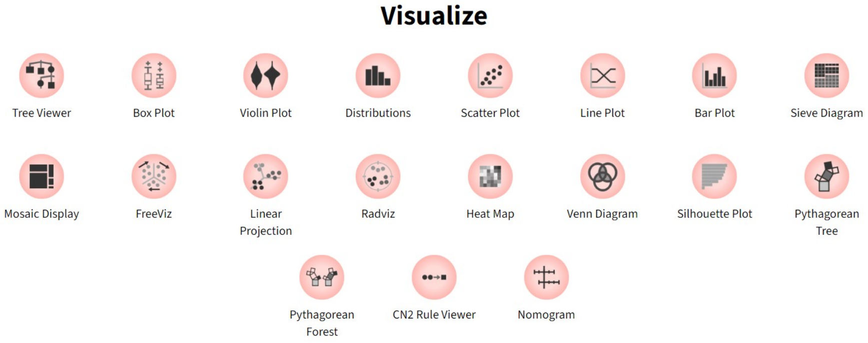


FIGURE 2.20: Visualize Widget

.

Model: for machine/deep learning models

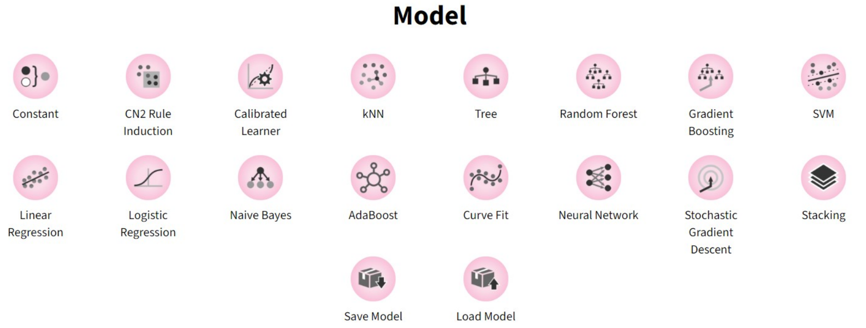


FIGURE 2.21: Model Widget

.

Evaluate: for evaluating the developed machine/deep learning model

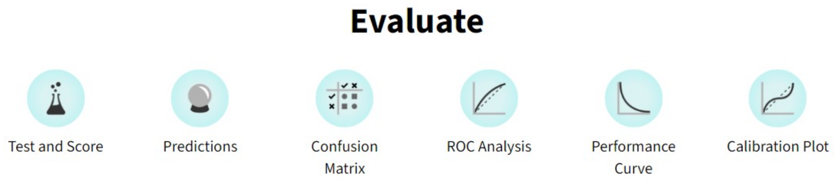


FIGURE 2.22: Evaluate Widget

.

Unsupervised: includes models/techniques for an unsupervised approach

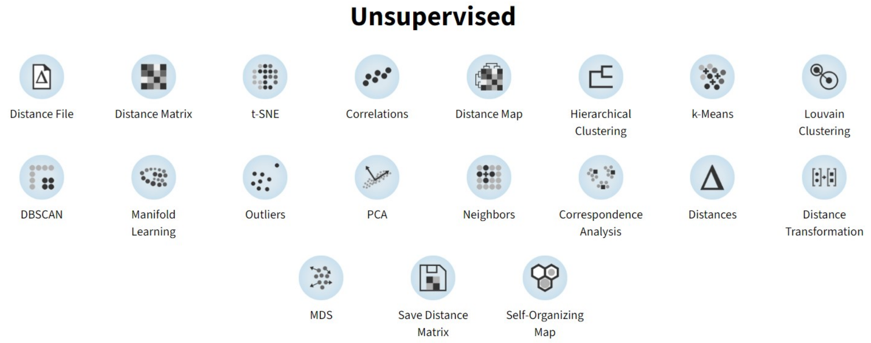


FIGURE 2.23: Unsupervised Widget

those Widgets are installed with default Orange, the user can add many other if he needs like Geo, image Analytics, network text...etc.

2.6 Conclusion

After the description of each data exploration tool, the next chapter will present a comparative analysis of these tools. We will carefully give a deep understanding of how these tools work internally. This will help us to know exactly what each tool can do and how to use its different features well.

Chapter 3

Comparative study

3.1 Introduction

The comparative study provides valuable insights into the performance and capabilities of leading data mining software tools. The analysis covers a range of popular tools, including RapidMiner, Weka, Orange, and KNIME, highlighting their unique strengths and weaknesses. Factors such as predictive analytics, ease of use, data pipelining, and support for various data mining tasks are examined to help organizations make informed decisions in selecting the most suitable tool for their specific needs. Understanding the distinct features and capabilities of these data mining tools is crucial for maximizing the benefits of data-driven insights and decision-making.

Comparative research aims to study the static and dynamic properties of different data mining tools.

3.2 An Overview of the contribution

Our contribution tends to carry out a comparative study of four popular free data mining tools (Rapid Minder, Weka, Orange, and KNIME) and analyses their strengths and weaknesses. The study highlights some salient features (performance optimisation, tasks, types of dataset, real-time data analysis, cost, language binding etc.) of these tools.

In this context, our study is organized around two main research axes:

- **Static Analysis** mostly focuses on several factors, including functionality, algorithms, costs, etc.
- **Dynamic Analysis** involves the performance study in different steps of data mining such as preprocessing, clustering and classification under various kinds of algorithms.

Figure 3.1 illustrates the overview of our research.

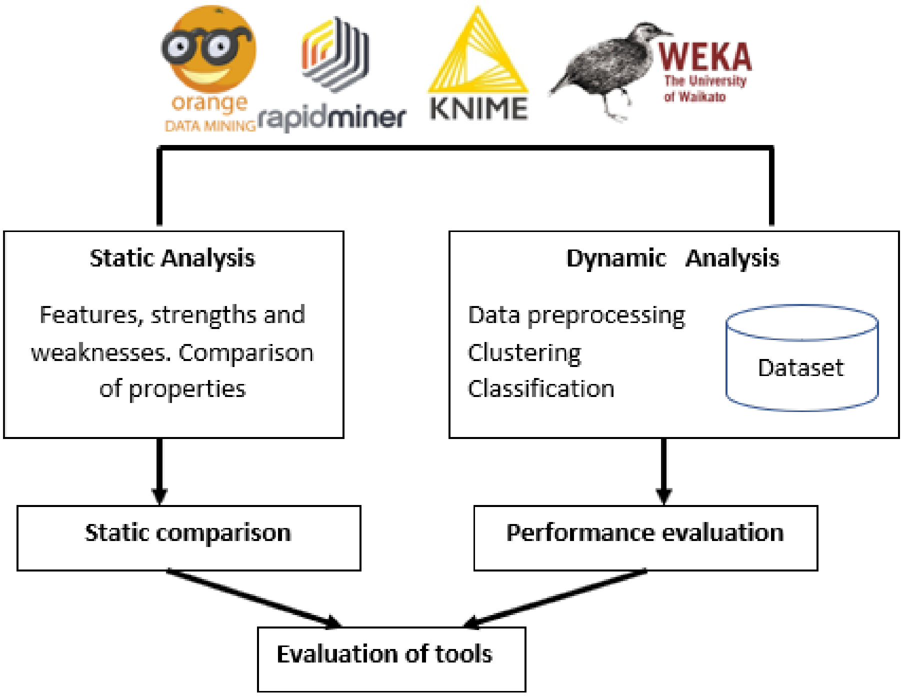


FIGURE 3.1: Overview of our contribution

3.3 Static analysis

Static analysis of data mining tools include the examination of code, data flows, and settings without running them in order to detect probable flaws, faults, and inefficiencies. This method is essential for assuring the strength, dependability, and effectiveness of data mining operations. The static analysis aims to analyse the structure and dependencies of tools such as RapidMiner, WEKA, Orange, and KNIME.

In the static analysis of data mining tools, several measures and criteria are typically considered to evaluate their performance, usability, and suitability for specific tasks.

3.3.1 Features and Functionalities

Each of these popular data mining tools offers unique features and functionalities that cater to a wide range of data analysis needs. Here's a detailed overview of what each tool provides:

- RapidMiner: This tool has many features like data prep, modeling, validation, and deployment. It uses a drag-and-drop way to make workflows.
- WEKA: Provides a comprehensive collection of machine learning algorithms for data preprocessing, classification, regression, clustering, association rules, and visualization.
- Orange: Focuses on data visualization and machine learning with a user-friendly interface. It offers a wide range of algorithms and tools for data exploration and modeling.
- KNIME: Offers a modular data analytics platform with a large number of ready-to-use components for data preprocessing, analysis, modeling, and reporting.

3.3.2 Ease of Use

- RapidMiner: is a machine learning tool. It has many features like data preprocessing, modeling, and deployment. It uses a drag-and-drop design to make workflows.
- WEKA: gives many machine learning algorithms. It preps data, classifies, predicts, groups, finds patterns, and visualizes.
- Orange: focuses on data viz and machine learning. It has a simple design. It gives algorithms and tools to explore and model data.
- KNIME: is a modular data platform. It has many ready parts for data preprocessing, analysis, modeling, and reporting.

3.3.3 Community Support and Documentation

- RapidMiner, WEKA, and KNIME have big groups of people that use them. These groups give information and help to users on websites and in papers. They have tutorials and notes that explain how to use the programs.

- Orange also has a group that helps people use it. But the group for Orange may be smaller than the groups for the other programs.

3.3.4 Integration and Extensibility

- KNIME and RapidMiner are known for their strong integration capabilities with other tools and platforms, allowing users to leverage a wide range of data sources and technologies.
- Orange and WEKA also offer some level of extensibility through plugins and APIs, but they may not be as extensive as RapidMiner and KNIME.

3.3.5 Summary

In summary, RapidMiner, WEKA, Orange, and KNIME each offer distinct features and functionalities tailored to various aspects of data mining and machine learning. These tools provide robust environments for data scientists and analysts to perform comprehensive data analysis, model building, and deployment.

Table 3.1 summarised the principal criteria, which are typically considered to evaluate the performance of different data mining tools.

TABLE 3.1: Summary of the static comparison

Feature	**RapidMiner**	**Weka**	**Orange**	**KNIME**
Main Feature	Visual workflow	Machine learning algorithms	Interactive data exploration	Data integration & manipulation
Strength	User-friendly, extensive pre-built components	Powerful algorithms, data visualization	Interactive exploration, scripting	Data integration, scripting
Weakness	Can be resource-intensive for complex workflows	Limited user interface customization	Limited support for distributed computing	Programming skills required for advanced tasks
Cost	Free and CommercIFl Versions, free version support only 10.000 row	Free	Free	Free
Programming Language	Java	Java	Python	Java
Supported Data Formats	CSV, XML, JSON, others	CSV, ARFF, others	Various, including Python data structures	Various
Machine Learning Algorithms	Extensive library	Comprehensive algorithms, including deep learning	Various, primarily supervised learning	Various, including deep learning

Feature	**RapidMiner**	**Weka**	**Orange**	**KNIME**
Data Visualization	Basic visualization components	Powerful charting and data exploration tools	Interactive visualizations	Extensive visualization capabilities
Real-time Data Analysis	Limited support	Limited support	Limited support	Some support through extensions
OSs	Windows macOS Lunix open source	Windows macOS Lunix open source	Windows macOS Lunix open source	Windows macOS Lunix open source
Tutoriels	video, Academy documents, Simple datasets offred	Youtube, Textbook and free coureses	Vidios, Free data mining coures , Educational blog	Community forum, Learning hub
Quality	Powerful and versatile with an advantage especIFlly in predictive analytics	Many classification methods	Creates particularly attractive and interesting data visualizations without extensive prior knowledge	Main open data mining tool that predictive analytics has made available to the general public

3.4 Dynamic analysis

In this dynamic study, we will evaluate RapidMiner, Weka, Orange, and KNIME by applying the data mining process to each tool for a specific task and comparing their performance. In this context, we use two datasets: a labelled dataset "Bank-additional-full.csv" and an unlabeled dataset "UCI-Credit-Card.csv", which are available in the Kaggle framework: https://www.kaggle.com/.

3.4.1 Dataset description

Dataset1 "Bank-additional-full"

Bank marketing campaigns dataset analysis Opening a Term Deposit. It is a dataset that describes Portugal's bank marketing campaign results. Conducted campaigns were based mostly on direct phone calls, offering bank clients to place a term deposit. If after all marking efforts client had agreed to place the deposit target variable marked 'yes', otherwise 'no'

Source of the data: https://archive.ics.uci.edu/ml/datasets/bank+marketing

Citation Request: This dataset is public available for research. The details are described in S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing.

Number of Instances: 41188

Number of Attributes: 21 attributes.

Some attribute information:

- campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

- pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

- cons.price.idx: consumer price index - monthly indicator (numeric)

Dataset2 "UCI-Credit-Card" contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

Source of the data: https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

Number of Instances: 30000

Number of Attributes: 25 attributes.

Some attribute information:

- LIMIT-BAL: Amount of given credit in NT dollars (includes individual and family/ supplementary credit.

- PAY-0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, . . . 8=payment delay for eight months, 9=payment delay for nine months and above).

- default.payment.next.month: Default payment.

Table 3.2 presents some properties of these datasets.

TABLE 3.2: Datasets Overview

Dataset	Dimension	Instances	Numeric datatype	Nominal datatype
Dataset1	21	41188	10	11
Dataset2	25	30000	25	0

3.4.2 Data Mining Using the RapidMiner Tool

1. Preprocessing phase

1-1- Importing the CSV File

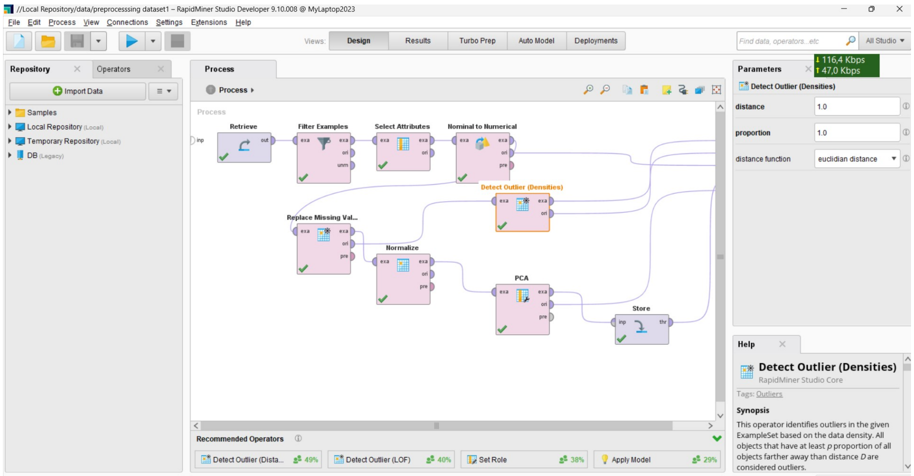
- Open RapidMiner.

- Import Data: In the top-left corner, click on "Repository" to access our local repository. Then, navigate to the location where we want to import the CSV dataset.

- Right-Click: Right-click on the folder where we want to import the dataset.
- Select Import: From the context menu, select "Import Data" and then choose "From File...".
- Browse for CSV File: A file browser window will pop up. Navigate to the location of the CSV dataset on the local machine, select it, and click "Open".
- Define features (Optional): RapidMiner will attempt to automatically detect the features (columns) and their types in our dataset. We can review and modify this if needed.
- Finish Importing: Once we have verified the features, click "Finish" to import the dataset into RapidMiner.
- Verify Dataset: After importing, we can verify that the dataset has been successfully imported by checking the folder we imported it into.

1-2-Preprocessing Steps

- Retrieve: to load the dataset.
- Filter examples: to select data to remove missing values.
- select features: to select columns of the dataset to use.
- Nominal to Numeric: to change string to integer.
- Replaces missing value: to replace the missing values if they exist.
- Normalize: normalize the data set for preparing for the next step.
- PCA: to apply principal components analysis on normalized data.
- Store: to store the dataset for the next steps.


//Local Repository/data/preprocessing dataset1 – RapidMiner Studio Developer 9.10.008 @ MyLaptop2023
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Repository
Operators
Import Data
Samples
Local Repository (Local)
Temporary Repository (Local)
DB (Legacy)
Process
Retrieve
Filter Examples
Select Attributes
Nominal to Numerical
Detect Outlier (Densities)
Replace Missing Val...
Normalize
PCA
Store
Recommended Operators
Detect Outlier (Dista... 49%
Detect Outlier (LOF) 40%
Set Role 38%
Apply Model 29%
116,4 Kbps
47,0 Kbps
Parameters
Help

Detect Outlier (Densities)

distance: 1.0

proportion: 1.0

distance function: euclidian distance

Detect Outlier (Densities)
RapidMiner Studio Core
Tags: Outliers

Synopsis

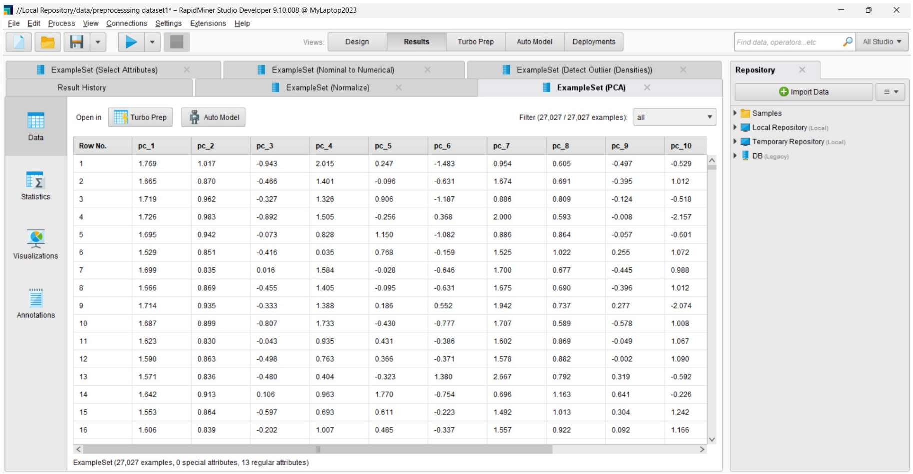
This operator identifies outliers in the given ExampleSet based on the data density. All objects that have at least *p* proportion of all objects farther away than distance *D* are considered outliers.

FIGURE 3.2: Preprocessing of the dataset 1

1-3-Executing the Workflow

- Click the "Run" (blue play icon) on the top toolbar.
- RapidMiner will execute the workflow, performing the preprocessing steps we defined.

1-4 Results: After executing the process, RapidMiner will open a new window displaying the results of each operator used in the process.


//Local Repository/data/preprocessing dataset1* – RapidMiner Studio Developer 9.10.008 @ MyLaptop2023
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
ExampleSet (Select Attributes)
ExampleSet (Nominal to Numerical)
ExampleSet (Detect Outlier (Densities))
Result History
ExampleSet (Normalize)
ExampleSet (PCA)
Data
Statistics
Visualizations
Annotations
Open in
Turbo Prep
Auto Model
Repository
Import Data
Samples
Local Repository (Local)
Temporary Repository (Local)
DB (Legacy)

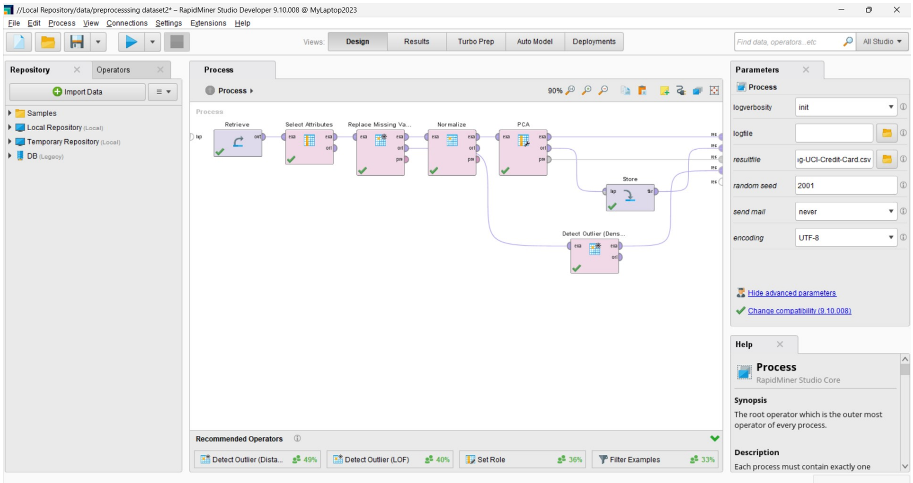
Filter (27,027 / 27,027 examples): all

Row No.	pc_1	pc_2	pc_3	pc_4	pc_5	pc_6	pc_7	pc_8	pc_9	pc_10
1	1.769	1.017	-0.943	2.015	0.247	-1.483	0.954	0.605	-0.497	-0.529
2	1.665	0.870	-0.466	1.401	-0.096	-0.631	1.674	0.691	-0.395	1.012
3	1.719	0.962	-0.327	1.326	0.906	-1.187	0.886	0.809	-0.124	-0.518
4	1.726	0.983	-0.892	1.505	-0.256	0.368	2.000	0.593	-0.008	-2.157
5	1.695	0.942	-0.073	0.828	1.150	-1.082	0.886	0.864	-0.057	-0.601
6	1.529	0.851	-0.416	0.035	0.768	-0.159	1.525	1.022	0.255	1.072
7	1.699	0.835	0.016	1.584	-0.028	-0.646	1.700	0.677	-0.445	0.988
8	1.666	0.869	-0.455	1.405	-0.095	-0.631	1.675	0.690	-0.396	1.012
9	1.714	0.935	-0.333	1.388	0.186	0.552	1.942	0.737	0.277	-2.074
10	1.687	0.899	-0.807	1.733	-0.430	-0.777	1.707	0.589	-0.578	1.008
11	1.623	0.830	-0.043	0.935	0.431	-0.386	1.602	0.869	-0.049	1.067
12	1.590	0.863	-0.498	0.763	0.366	-0.371	1.578	0.882	-0.002	1.090
13	1.571	0.836	-0.480	0.404	-0.323	1.380	2.667	0.792	0.319	-0.592
14	1.642	0.913	0.106	0.963	1.770	-0.754	0.696	1.163	0.641	-0.226
15	1.553	0.864	-0.597	0.693	0.611	-0.223	1.492	1.013	0.304	1.242
16	1.606	0.839	-0.202	1.007	0.485	-0.337	1.557	0.922	0.092	1.166

ExampleSet (27,027 examples, 0 special attributes, 13 regular attributes)

FIGURE 3.3: Result of the preprocessing of dataset1

For the second dataset, we follow the same steps but omit certain operators, such as NominalToNumeric and Filter, because the dataset is numeric and contains no missing values.


//Local Repository/data/preprocesssing dataset2* – RapidMiner Studio Developer 9.10.008 @ MyLaptop2023

Parameters: Process

logverbosity: init

logfile:

resultfile: ıg-UCI-Credit-Card.csv

random seed: 2001

send mail: never

encoding: UTF-8

Help: Process (RapidMiner Studio Core)

Synopsis: The root operator which is the outer most operator of every process.

Description: Each process must contain exactly one

FIGURE 3.4: Result of preprocessing of dataset 2

At the end of the preprocessing for both datasets, we obtain the following results, where: II: initial items, FI: Final Items, IF: Initial features, FaPCA: features after PCA transformation, MV missing Values.

TABLE 3.3: Preprocessing Result

Dataset	II	FI	IF	Type	FaPCA	MV
Dataset1	41118	27027	21	nominal	16	14091
Dataset2	30000	30000	25	numeric	23	0

2. Clustering: After preprocessing, navigate to the clustering section of the tool. Here, we will select clustering as the task to perform.

2.1- Auto Model

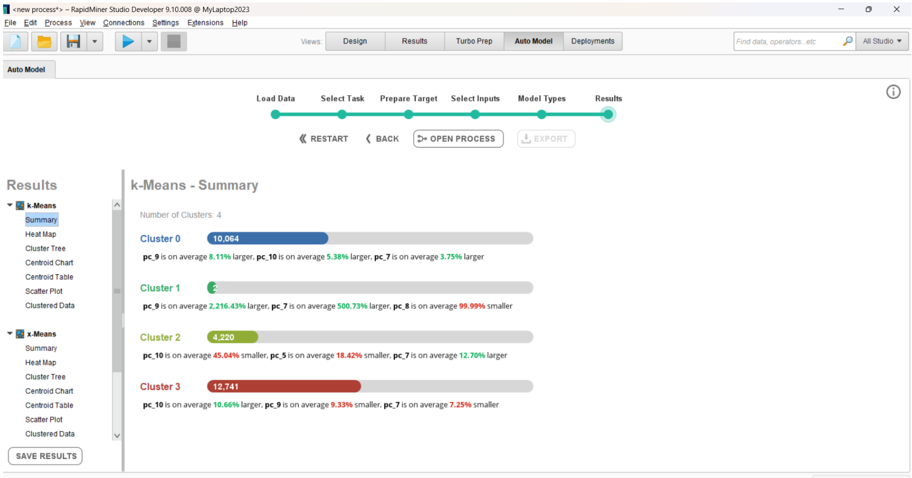
Algorithm Selection Specify that we want to use both X-means and K-means algorithms for clustering. We should have options to select these algorithms from a dropdown menu or similar interface.

Set Parameters For each algorithm (X-means and K-means), we will need to fix the value of the k parameter. This determines the number of clusters the algorithm will identify.

Run Once we have set the parameters for both algorithms, initiate the clustering process by clicking on the "run" button. The Auto Model tool will execute the algorithms on our dataset.

View Results After the clustering process completes, we should be able to view the results. This may include visualizations such as cluster plots or cluster summaries, which show how the data points are grouped into clusters by each algorithm.

Figure 3.5 shows the results of clustering of the labelled dataset1 after removing the target column. The goal of this task is to compare the results of clustering with the original labelled dataset.



<new process*> – RapidMiner Studio Developer 9.10.008 @ MyLaptop2023

k-Means - Summary

Number of Clusters: 4

Cluster 0 10,064

pc_9 is on average 8.11% larger, **pc_10** is on average 5.38% larger, **pc_7** is on average 3.75% larger

Cluster 1 2

pc_9 is on average 2,216.43% larger, **pc_7** is on average 500.73% larger, **pc_8** is on average 99.99% smaller

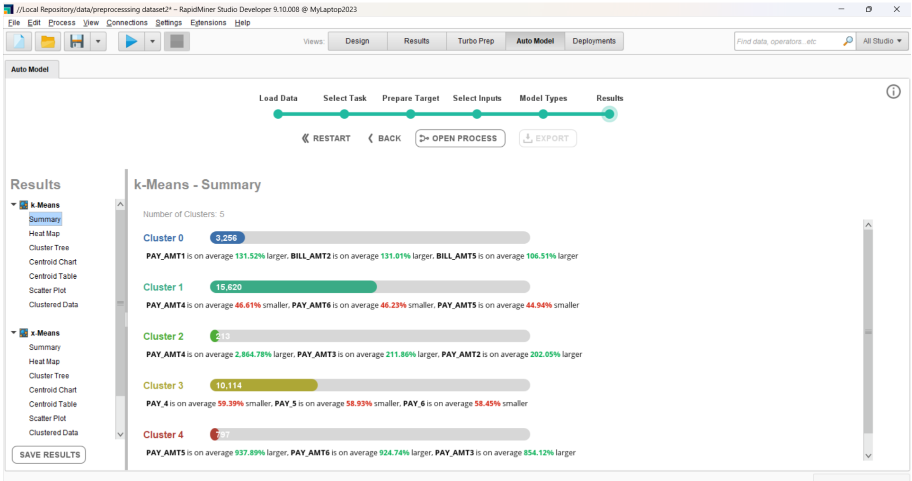
Cluster 2 4,220

pc_10 is on average 45.04% smaller, **pc_5** is on average 18.42% smaller, **pc_7** is on average 12.70% larger

Cluster 3 12,741

pc_10 is on average 10.66% larger, **pc_9** is on average 9.33% smaller, **pc_7** is on average 7.25% smaller

FIGURE 3.5: Results K-means on dataset 1



k-Means - Summary

Number of Clusters: 5

Cluster 0 3,256

PAY_AMT1 is on average 131.52% larger, BILL_AMT2 is on average 131.01% larger, BILL_AMT5 is on average 106.51% larger

Cluster 1 15,620

PAY_AMT4 is on average 46.61% smaller, PAY_AMT6 is on average 46.23% smaller, PAY_AMT5 is on average 44.94% smaller

Cluster 2 213

PAY_AMT4 is on average 2,864.78% larger, PAY_AMT3 is on average 211.86% larger, PAY_AMT2 is on average 202.05% larger

Cluster 3 10,114

PAY_4 is on average 59.39% smaller, PAY_5 is on average 58.93% smaller, PAY_6 is on average 58.45% smaller

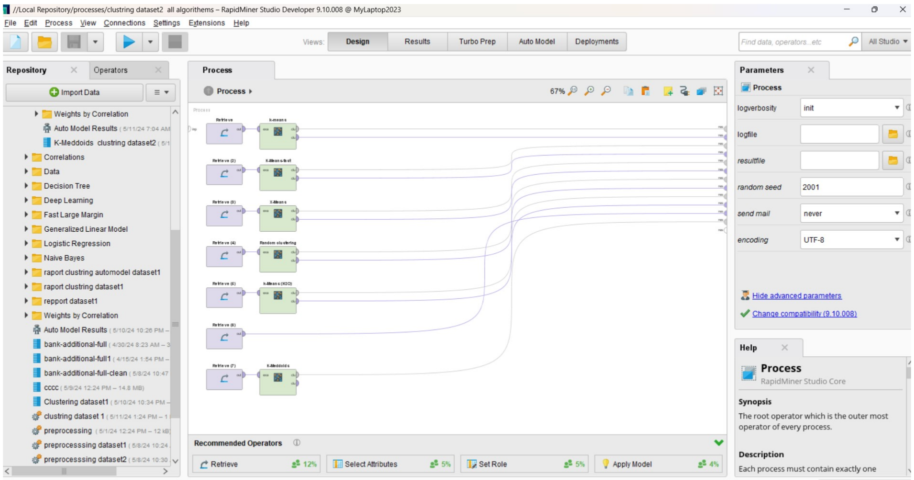
Cluster 4 797

PAY_AMT5 is on average 937.89% larger, PAY_AMT6 is on average 924.74% larger, PAY_AMT3 is on average 854.12% larger

FIGURE 3.6: Results K-means on dataset 2

2.2- Other algorithms

In addition, there are other algorithms that we will clustering compare their results. We can cluster reprocessed datasets by making a model like the next figure


//Local Repository/processes/clustring dataset2 all algorithems – RapidMiner Studio Developer 9.10.008 @ MyLaptop2023
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Repository
Operators
Import Data
Weights by Correlation
Auto Model Results (5/11/24 7:04 AM
K-Meddoids clustring dataset2 (5/1
Correlations
Data
Decision Tree
Deep Learning
Fast Large Margin
Generalized Linear Model
Logistic Regression
Naive Bayes
raport clustring automodel dataset1
raport clustring dataset1
repport dataset1
Weights by Correlation
Auto Model Results (5/10/24 10:26 PM –
bank-additional-full (4/30/24 8:23 AM – 3
bank-additional-full1 (4/15/24 1:54 PM –
bank-additional-full-clean (5/8/24 10:47
cccc (5/9/24 12:24 PM – 14.8 MB)
Clustering dataset1 (5/10/24 10:34 PM –
clustring dataset 1 (5/11/24 1:24 PM – 1
preprocessing (5/1/24 12:24 PM – 12 kB)
preprocesssing dataset1 (5/8/24 10:24
preprocesssing dataset2 (5/8/24 10:30
Process
Process
67%
Retrieve
k-means
Retrieve (2)
K-Means-fast
Retrieve (3)
X-Means
Retrieve (4)
Random clustering
Retrieve (5)
k-Means (H2O)
Retrieve (6)
Retrieve (7)
K-Meddoids
Recommended Operators
Retrieve 12%
Select Attributes 5%
Set Role 5%
Apply Model 4%
Parameters
Process
Hide advanced parameters
Change compatibility (9.10.008)
Help

logverbosity init

logfile

resultfile

random seed 2001

send mail never

encoding UTF-8

Process

RapidMiner Studio Core

Synopsis

The root operator which is the outer most operator of every process.

Description

Each process must contain exactly one

FIGURE 3.7: Clustering dataset with chosen Algorithms

TABLE 3.4: Clustering Results of Dataset1

Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
K-means-fast	17514 items	2 items	1191 items	1041 items	7279 items
K-Means	7279 items	2 items	1191 items	1039 items	17516 items
X-Means	17514 items	2 items	1191 items	1041 items	7279 items
K-Means (H2O)	4357 items	2 items	4045 items	15662 items	2961 items
K-Medoids	135 items	6076 items	17139 items	2614 items	1063 items
Random Clustering	5399 items	5470 items	5387 items	5465 items	5306 items

TABLE 3.5: Clustering Results of Dataset2

Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
K-means-fast	25346 items	1 item	1 item	4650 items	2 items
K-Means	15728 items	3 items	10606 items	3661 items	2 items
X-Means	25346 items	1 item	1 item	4650 items	2 items
K-Means (H2O)	10057 items	4 items	1164 items	15646 items	3129 items
K-Medoids	2474 items	4172 items	479 items	22348 items	527 items
Random Clustering	5972 items	6076 items	5984 items	6080 items	5888 items

3. classification

To classify datasets, we will use the Auto Model with its various algorithms. These algorithms can also perform classification. We will load the labelled dataset, choose the "Predict" option, and select the desired features to predict. Then, click "Next" and set "Map New Classes to New Values." Click "Next" again, choose the featuress, and proceed. Select all algorithms and activate them, then click "Run."

The following tables summarize the results obtained after prediction and classification.

Where: CE: Classification Error, TT: total Time, TT (1000 rows): Training Time (1,000 Rows), ST (1,000 Rows): scoring time (1000 rows).

TABLE 3.6: Performance Comparison of Various Models of Auto Model to classify dataset 1

Model	**CE**	**Standard DevIFtion**	**Gains**	**TT**	**TT (1,000 rows)**	**ST (1,000 lines)**
Naive Bayes	31.80%	±0.60%	3270.0	3s	6ms	203ms
Generalized Linear Model	18.53%	±0.43%	5296.0	33s	712ms	261ms
Logistic Regression	44.16%	±0.80%	1288.0	7s	76ms	477.52ms
Fast Large Margin	42.68%	±0.52%	1516.0	3min 59s	1ms	463ms
Deep Learning	20.78%	±0.23%	4938.0	15s	291ms	270.49ms
Decision Tree	33.66%	±0.69%	2968.0	5s	20ms	200ms
Random forest	19.9%	±0.7%	5082	1m5s	38ms	431ms
Gradient Boosted Trees	13.2%	±0.3%	6078.0	2m32s	38ms	770ms
Support Vector machine	28.6%	±0.5%	2092	51m16s	32s	24s

TABLE 3.7: Performance Comparison of Various Models of model to classify dataset2

Model	CE	Standard DevIFtion	Gains	Total Time	TT (1,000 Rows)	ST (1,000 Rows)
Naive Bayes	68.4%	(±) 0.6%	-1664	8s	9ms	528ms
Generalized Linear Model	24.3%	(±) 0.3%	5966	35s	632ms	293ms
Logistic Regression	24.6%	(±) 0.3%	5912	9s	55ms	739ms
Fast Large Margin	24.6%	(±) 0.4%	5918	2m0s	411s	698ms
Deep Learning	26.6%	(±) 0.4%	5636	7s	29ms	253ms
Random forest	25.3%	(±) 0.7%	5764	1m28s	93ms	1s
Gradient Boosted Trees	22.3%	(±) 0.3%	6272	4m13s	382ms	838ms
Support Vector machine	15.9 %	(±) 0.6%	3640	9m 3s	749ms	2s

3.4.3 Data Mining Using the Weka Tool

1. Preprocessing phase

1.1-Importing the CVS file

- Open Weka: Launch the Weka application on our computer.
- Open the Explorer: Once Weka is running, open the Weka Explorer interface.
- Load Data: In the Explorer interface, go to the "Preprocess" tab.
- Load CSV File: Click on the "Open file" button then select "Open".
- Choose CSV Loader: Choose "CSV Loader" from the list of available loaders.
- Preview Data: Weka will display a preview of our data. We make sure it looks correct before proceeding.
- Save ARFF File (Optional): Weka usually works with ARFF (features-Relation File Format) files.

1.2-Preprocessing Steps

For the first dataset **bank-additional-full** In process canvas: we place"retrieve" operator and set it's data source to the dataset previous loaded. We do the same previous steps did in Rapidminer:

To remove missing values using the "ReplaceMissingValues" filter in WEKA:

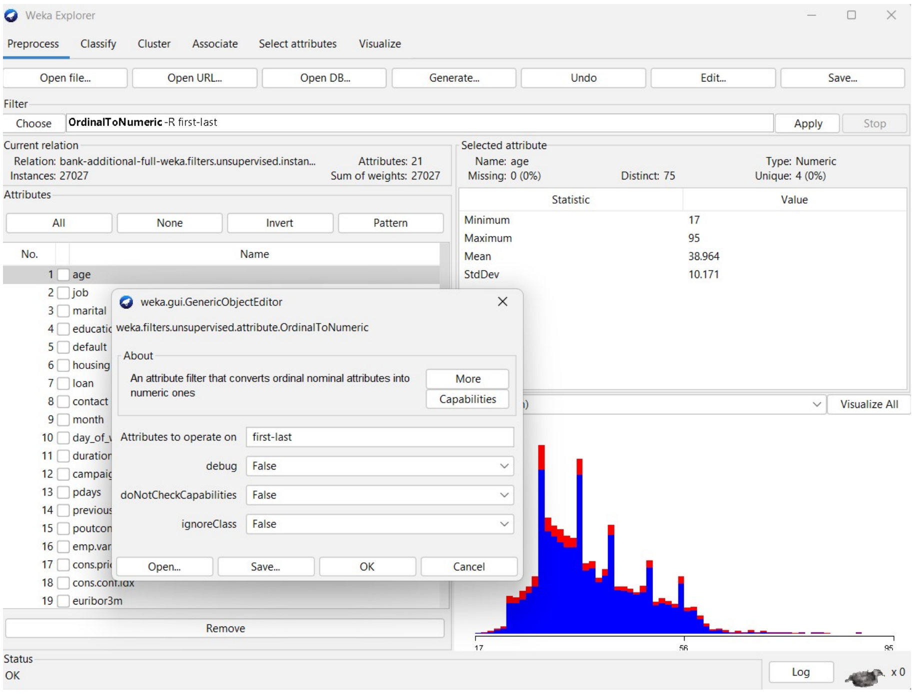
1. Go to the "Preprocess" tab.
2. Choose the "Filter" option.
3. Select "unsupervised.features.ReplaceMissingValues" from the list of filters.
4. Alternatively, you can also use the "NumericCleaner" filter to remove instances with missing values in numeric featuress.
5. Click on the filter and configure it according to your needs. Apply the filter.
6. Save the result to use it in the next step.

Convert nominal "labels" to "numeric binary value": One frequent preprocessing step in WEKA is to convert nominal properties to numerical ones, especIFlly when working with machine learning algorithms that need numerical input. In practice, replace step 3 by: "supervised.features.OrdinalToBinary" from the list of filters to convert nominal featuress to binary featuress and chose all featuress. Then apply the Filter.

In order to centralize a dataset in WEKA, it is customary to deduct the mean of each features from all the corresponding features values. This method involves the adjustment of the data so that its mean value is zero. Here is how we can do using the dataset preprocessing :

- Choose the "Filter" option.
- Select "unsupervised.features.Center" from the list of filters.
- This filter centers the data by subtracting the mean of each
- features from all the values of that features.
- Apply the filter.

For the second dataset **UCI-Credit-Card** we do the same steps without "NominalToBinary" because all the featuress are Numerical.



OrdinalToNumeric -R first-last

Relation: bank-additional-full-weka.filters.unsupervised.instan... Attributes: 21
Instances: 27027 Sum of weights: 27027

No.	Name
1	☐ age
2	☐ job
3	☐ marital
4	☐ educatic
5	☐ default
6	☐ housing
7	☐ loan
8	☐ contact
9	☐ month
10	☐ day_of_\
11	☐ duratior
12	☐ campaig
13	☐ pdays
14	☐ previous
15	☐ poutcon
16	☐ emp.var
17	☐ cons.pri
18	☐ cons.conf.idx
19	☐ euribor3m

Name: age Type: Numeric
Missing: 0 (0%) Distinct: 75 Unique: 4 (0%)

Statistic	Value
Minimum	17
Maximum	95
Mean	38.964
StdDev	10.171

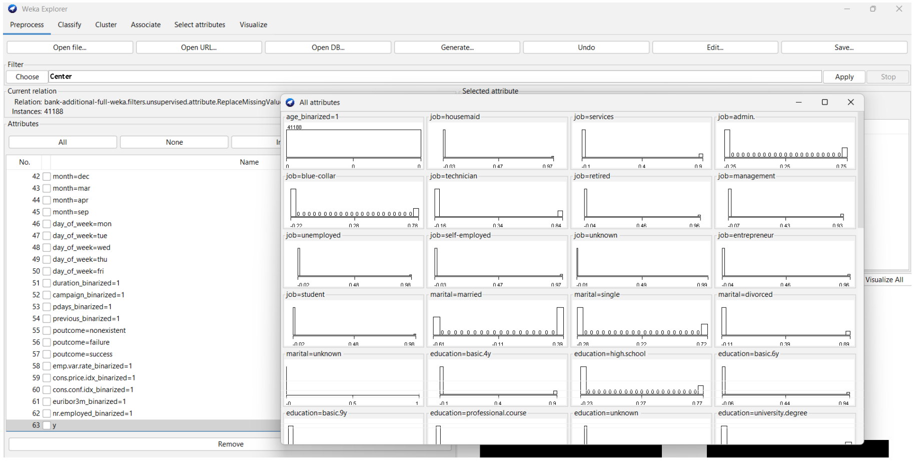
weka.filters.unsupervised.attribute.OrdinalToNumeric

An attribute filter that converts ordinal nominal attributes into numeric ones

Attributes to operate on: first-last
debug: False
doNotCheckCapabilities: False
ignoreClass: False

Status
OK

FIGURE 3.8: Nominal to binary filter



Filter

Choose Center

Current relation

Relation: bank-additional-full-weka.filters.unsupervised.attribute.ReplaceMissingValu

Instances: 41188

Attributes

No.		Name
42	□	month=dec
43	□	month=mar
44	□	month=apr
45	□	month=sep
46	□	day_of_week=mon
47	□	day_of_week=tue
48	□	day_of_week=wed
49	□	day_of_week=thu
50	□	day_of_week=fri
51	□	duration_binarized=1
52	□	campaign_binarized=1
53	□	pdays_binarized=1
54	□	previous_binarized=1
55	□	poutcome=nonexistent
56	□	poutcome=failure
57	□	poutcome=success
58	□	emp.var.rate_binarized=1
59	□	cons.price.idx_binarized=1
60	□	cons.conf.idx_binarized=1
61	□	euribor3m_binarized=1
62	□	nr.employed_binarized=1
63	□	y

FIGURE 3.9: Normalize dataset using "Centre" filter

Applying Principal Component Analysis (PCA) in Weka is possible with the "PrincipalComponents" filter for dimensionality reduction.

Launch Weka and use the "Explorer" interface to load the dataset, which is the outcome of the "cleaning" and "centralize" procedures.

Apply PCA:

- Go to the "Preprocess" tab in the Weka Explorer interface.
- Select the "PrincipalComponents" filter from the list of filters.
- Click on the filter to open its configuration window.

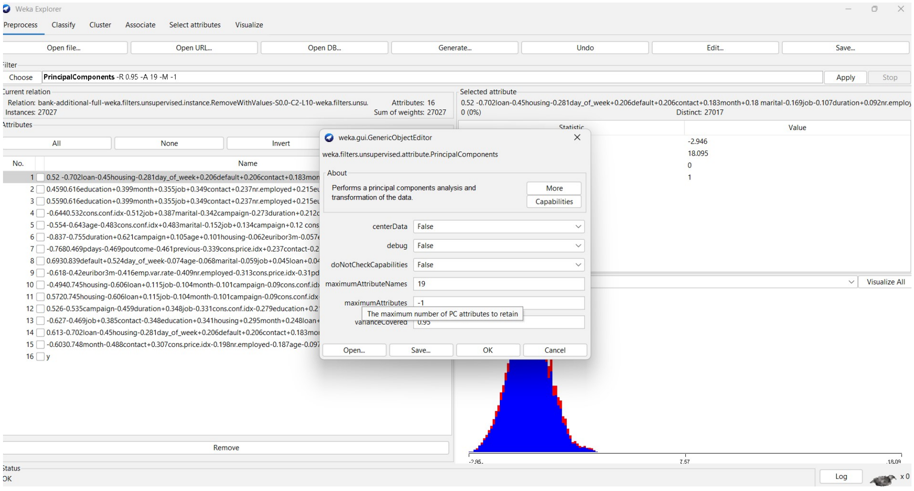
Configure PCA: In the PrincipalComponents filter configuration window, we can specify various parameters:

- We Choose the number of principal components to retain (e.g. if we want to reduce the dimensionality of our data).
- We can also choose whether to standardize the data before applying PCA.

Apply Filter: After configuring the PCA filter, We click on the "Apply" button to apply the filter to the dataset.

Evaluate the Transformed Data

- Once PCA is applied, we will get a new dataset with the transformed features (principal components).
- we can now use this transformed dataset for further analysis or modeling.
- We can visualize the transformed data using techniques such as scatter plots or histograms to understand the distribution of principal components.



Filter
Choose **PrincipalComponents** -R 0.95 -A 19 -M -1

Current relation
Relation: bank-additional-full-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C2-L10-weka.filters.unsu.
Instances: 27027
Attributes: 16
Sum of weights: 27027

Attributes

No.		Name
1	□	0.52 -0.702loan-0.45housing-0.281day_of_week+0.206default+0.206contact+0.183mon
2	□	0.4590.616education+0.399month+0.355job+0.349contact+0.237nr.employed+0.215eu
3	□	0.5590.616education+0.399month+0.355job+0.349contact+0.237nr.employed+0.215eu
4	□	-0.6440.532cons.conf.idx-0.512job+0.387marital-0.342campaign-0.273duration+0.212c
5	□	-0.554-0.643age-0.483cons.conf.idx+0.483marital-0.152job+0.134campaign+0.12 cons
6	□	-0.837-0.755duration+0.621campaign+0.105age+0.101housing-0.062euribor3m-0.057e
7	□	-0.7680.469pdays-0.469poutcome-0.461previous-0.339cons.price.idx+0.237contact-0.2
8	□	0.6930.839default+0.524day_of_week-0.074age-0.068marital-0.059job+0.045loan+0.04
9	□	-0.618-0.42euribor3m-0.416emp.var.rate-0.409nr.employed-0.313cons.price.idx-0.31pd
10	□	-0.4940.745housing-0.606loan+0.115job-0.104month-0.101campaign-0.09cons.conf.idx
11	□	0.5720.745housing-0.606loan+0.115job-0.104month-0.101campaign-0.09cons.conf.idx
12	□	0.526-0.535campaign-0.459duration+0.348job-0.331cons.conf.idx-0.279education+0.2
13	□	-0.627-0.469job+0.385contact-0.348education+0.341housing+0.295month+0.248loan+
14	□	0.613-0.702loan-0.45housing-0.281day_of_week+0.206default+0.206contact+0.183mor
15	□	-0.6030.748month-0.488contact+0.307cons.price.idx-0.198nr.employed-0.187age-0.097
16	□	y

Selected attribute
0.52 -0.702loan-0.45housing-0.281day_of_week+0.206default+0.206contact+0.183month+0.18 marital-0.169job-0.107duration+0.092nr.employ
0 (0%)
Distinct: 27017

Statistic	Value
	-2.946
	18.095
	0
	1

weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.PrincipalComponents
About
Performs a principal components analysis and transformation of the data.
More
Capabilities
centerData False
debug False
doNotCheckCapabilities False
maximumAttributeNames 19
maximumAttributes -1
The maximum number of PC attributes to retain
varianceCovered 0.95
Open... Save... OK Cancel

Status
OK

FIGURE 3.10: Result of PCA applied on dataset1

At the end of the preprocessing of the two datasets, we get this result

TABLE 3.8: Preprocessing Result

Dataset	II	FI	IF	Type	FaPCA	MV
Dataset1	41118	27027	21	NOM+NUM	16	14091
Dataset2	30000	30000	25	NOM+NUM	17	0

Where: II:initial items, FI: Final items; IF:Initial features,FaPCA: features after PCA transformation, M V: Missing Values.

2-Clustering dataset: to cluster dataset in Weka We can apply the following steps:

- Open WEKA.
- Go to the "Explorer" interface in WEKA.
- Click on the "Open file" button and select your dataset file. Once loaded, our dataset will appear in the "Preprocess" panel.
- click on cluster and choose and Configure the parameters of the selected clustering algorithm, such as the number of clusters (k-means),
- After setting up the cluster and its options, we click on the "Start" button to execute the clustering process. Item WEKA will perform clustering on the dataset using the selected algorithm and parameters.
- Once the clustering process is complete, you can view the results in the "Cluster Assignments" panel.
- We will utilize the results of the PCA application on datasets.

After the clustering process is complete, we should be able to view the results.

TABLE 3.9: Clustering Results of Dataset1

Algorithm	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**time**
EM	6557	3572	1806	12423	2669	2.07s
Canopy	12600	4838	5900	2512	1177	0.07s
FarthestFirst	21361	588	557	2565	1956	0.03 s
Filterd-Clusterer	44178	8866	1187	9801	2985	0.25 s
make density base clustering	4186	8417	1191)	10247	9286	0.32 s
Simple-KMeans	4178	8866	1187	9801	2995	0.25 s

TABLE 3.10: Clustering Results of Dataset2

Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	time
EM	1526	366	21747	4822	1539	5.67s
Canopy	16968	6636	18	27	6351	0.1s
FarthestFirst	23359	1	3	1	6636	0.03s
Filterd-Clusterer	9530	7065	2449	4187	6769	0.41 s
make density base clustering	9005	6829	2574	4347	7245	0.48 s
Simple-KMeans	9530	7065	2449	4187	6769	0.25 s

3- Classification: Before we address classification, we must do an important step: add the clusters to the preprocessed dataset.

- Open WEKA.
- click on Explorer then preprocessing and load dataset preprocessed.
- click on filter-choose -unsupervised-features-Addcluster.
- configure the setting and apply.

Now the labels of clusters are added to the dataset and we can perform classification.

Dataset1 classification Results

TABLE 3.11: weka classifiers bayes NaiveBayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.927	0.012	0.934	0.927	0.931	0.918	0.979	0.958	cluster1
0.959	0.001	0.995	0.959	0.977	0.972	0.991	0.983	cluster2
0.995	0.025	0.861	0.995	0.923	0.913	0.993	0.946	cluster3
0.989	0.000	1.000	0.989	0.994	0.992	1.000	1.000	cluster4
0.932	0.007	0.975	0.932	0.953	0.940	0.980	0.970	cluster5

Build the model in 0.09 seconds.

TABLE 3.12: weka classifiers trees RandomForest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.991	0.002	0.991	0.991	0.991	0.989	1.000	1.000	cluster1
0.994	0.001	0.994	0.994	0.994	0.993	1.000	1.000	cluster2
0.991	0.002	0.985	0.991	0.988	0.986	1.000	0.999	cluster3
0.999	0.000	1.000	0.999	0.999	0.999	1.000	1.000	cluster4
0.994	0.001	0.996	0.994	0.995	0.993	1.000	1.000	cluster5

Build the model in 8.94 seconds.

TABLE 3.13: weka classifiers trees RandomTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.976	0.005	0.973	0.976	0.974	0.970	0.985	0.953	cluster1
0.986	0.003	0.988	0.986	0.987	0.984	0.992	0.976	cluster2
0.968	0.005	0.970	0.968	0.969	0.964	0.981	0.943	cluster3
0.999	0.001	0.998	0.999	0.998	0.997	0.999	0.997	cluster4
0.987	0.004	0.987	0.987	0.987	0.983	0.992	0.977	cluster5

Build the model in 0.13 seconds.

TABLE 3.14: weka classifiers rules DecisionTable

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.918	0.023	0.880	0.918	0.899	0.880	0.991	0.952	cluster1
0.985	0.002	0.990	0.985	0.987	0.985	0.999	0.998	cluster2
0.820	0.012	0.917	0.820	0.866	0.848	0.985	0.938	cluster3
0.996	0.009	0.979	0.996	0.988	0.982	0.999	0.998	cluster4
0.982	0.010	0.967	0.982	0.975	0.967	0.998	0.992	cluster5

Build the model in 3.02 seconds.

TABLE 3.15: weka classifiersr Rules JRip

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.994	0.001	0.995	0.994	0.994	0.993	1.000	0.998	cluster1
0.994	0.001	0.995	0.994	0.994	0.993	0.999	0.996	cluster2
0.993	0.001	0.992	0.993	0.993	0.992	0.999	0.997	cluster3
0.999	0.001	0.998	0.999	0.999	0.998	1.000	0.998	cluster4
0.996	0.001	0.997	0.996	0.996	0.995	1.000	0.999	cluster5

Build the model in 7.51 seconds.

TABLE 3.16: weka classifiers rules ZeroR

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.000	0.000	?	0.000	?	?	0.500	0.156	cluster1
0.000	0.000	?	0.000	?	?	0.500	0.177	cluster2
0.000	0.000	?	0.000	?	?	0.500	0.136	cluster3
1.000	1.000	0.296	1.000	0.457	?	0.500	0.296	cluster4
0.000	0.000	?	0.000	?	?	0.500	0.234	cluster5

Build the model in 0.02 seconds.

Dataset2 classification Results

TABLE 3.17: Weka classifiers Bayes NaiveBayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.838	0.019	0.900	0.838	0.868	0.843	0.977	0.923	cluster1
0.913	0.047	0.861	0.913	0.886	0.849	0.966	0.920	cluster2
0.879	0.013	0.892	0.879	0.886	0.872	0.983	0.886	cluster3
0.867	0.046	0.900	0.867	0.883	0.830	0.967	0.885	cluster4
0.896	0.032	0.841	0.896	0.868	0.843	0.963	0.929	cluster5

Build the model in 0.4 seconds.

TABLE 3.18: weka classifiers trees RandomForest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster1
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster2
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster3
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster4
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster5

Build the model in 1.42 seconds

TABLE 3.19: Weka classifiers Random Tree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster1
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster2
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster3
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster4
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	cluster5

Build the model in 0.02 seconds

TABLE 3.20: Weka classifiers DecisionTable

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.937	0.041	0.819	0.937	0.874	0.849	0.982	0.883	cluster1
0.930	0.054	0.846	0.930	0.886	0.849	0.979	0.909	cluster2
0.025	0.004	0.450	0.025	0.047	0.085	0.691	0.218	cluster3
0.949	0.088	0.837	0.949	0.889	0.835	0.970	0.910	cluster4
0.919	0.030	0.853	0.919	0.885	0.863	0.987	0.905	cluster5

Build the model in 0.04 seconds

TABLE 3.21: weka classifiersr Rules JRip

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.964	0.007	0.965	0.964	0.965	0.957	0.990	0.976	cluster1
0.966	0.006	0.980	0.966	0.973	0.964	0.994	0.982	cluster2
0.964	0.005	0.962	0.964	0.963	0.958	0.988	0.963	cluster3
0.981	0.014	0.971	0.981	0.976	0.965	0.991	0.969	cluster4
0.960	0.008	0.960	0.960	0.960	0.952	0.990	0.969	cluster5

Build the model in 0.03 seconds

TABLE 3.22: Weka classifiers Rule Zero

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.157	0.044	0.418	0.157	0.228	0.174	0.557	0.206	cluster1
0.894	0.165	0.635	0.894	0.743	0.659	0.865	0.593	cluster2
0.031	0.006	0.381	0.031	0.058	0.083	0.513	0.118	cluster3
0.903	0.333	0.564	0.903	0.694	0.533	0.785	0.540	cluster4
0.185	0.048	0.422	0.185	0.257	0.197	0.569	0.207	cluster5

Build the model in 0.011 seconds

3.4.4 Data Mining Using the Orange Tool

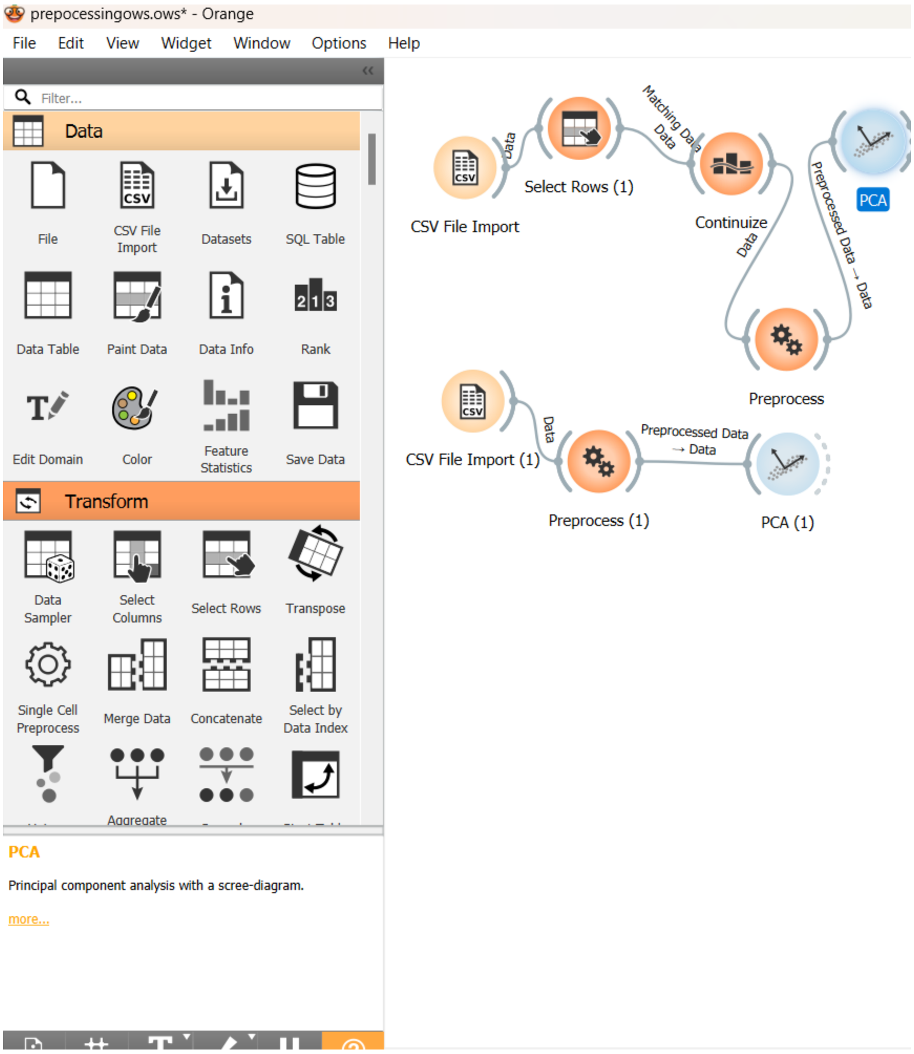
1-Preprocessing phase

1.1-Importing the CSV file

- Open the application.
- Drag and drop a "CSV import file" widget into the Canvas.
- Click on "CSV import file" widget and browse to the location file.
- Orange will automatically detect the file format and load it into the Canvas.

1.2-Preprocessing Steps

- Drag and drop "Select Rows, continue, Preprocessing, and PCA widgets into the canvas.
- Connect the input data of every one of those widgets to the output data of the previous widget.
- We use select rows to remove missing values, continue to convert nominal featuress to ordinal featuress, preprocess to centralise it and PCA to reduce the dimension.



PCA

Principal component analysis with a scree-diagram.

more...

FIGURE 3.11: preprocessing datasets in orange

At the end of the preprocessing on the two datasets, we get this result

TABLE 3.23: Preprocessing Result

Dataset	**II**	**FI**	**IF**	**Type**	**FaPCA**	**M V**
Dataset1	41118	27027	21	NOM+NUM	15	14091
Dataset2	30000	30000	25	NOM+NUM	17	0

Where: II: initial items, FI: Final Items, IF: Initial features, FaPCA: features after PCA transformation, MV missing Values.

2-Clustering dataset To cluster datasets in Orange, We will use the results of preprocessing and use the fourth algorithms unsupervised follow the next steps:

Hierarchical Clustering

- Drag and drop Distances and Hierarchical Clustering widgets in Orange canvas. Link the input of the distance widget to the output of the PCA widget and its output to the input of the Hierarchical Clustering widget.
- Configure the widget (e.g., choose distance metric and linkage method).

k-Means Clustering

- Drag and drop k-Means Clustering widget in Orange canvas. Link the input k-Means Clustering to the output of the PCA widget.
- Configure the k-Means Clustering widget (set the number of clusters).

DBSCAN Clustering

- Drag and drop the DBSCAN Clustering widget in Orange canvas. Link the input DBSCAN Clustering widget to the output of the PCA widget.
- Configure the widget (set epsilon and minimum samples)
- Configure the k-Means Clustering widget (set the number of clusters).

Interactive K-Means

- Drag and drop Interactive K-Means Clustering widget in Orange canvas .link the input Interactive K-Means widget to the output of PCA widget.
- Click on Interactive K-Means widget a graphics window will open here we can add or remove centre of cluster correspond to K-means k.

After the clustering process is complete, We should be able to view the results

TABLE 3.24: Clustering Results of Dataset1

Algorithm	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
Hierarchical	2 items	1190 items	3494 items	8077 items	14264 items
K-Means	4030 items	10169 items	1191 items	11622 items	2 items
DBSCAN	25685 items	1048 items	4 items	3 items	6 items
Interactive K-Means	9598 items	12153 items	4085 items	171 items	1020 items

TABLE 3.25: Clustering Results of Dataset2

Algorithm	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
Hierarchical	593 items	9224 items	98 items	82 items	20003 items
K-Means	6525 items	8811 items	6610 items	7733 items	294 items
DBSCAN	580 items	6721 items	12147 items	7817 items	2735 items
Interactive K-Means	29018 items	3 items	6 items	6 items	4 items

3-Classification To classify dataset with Orange data mining platform we feel this steps:

- Drag and drop a classification widget such as "Logistic Regression", "Random Forest", or "k-Nearest Neighbors" into the workspace.
- Connect the training data output from the "Data Sampler" to the chosen classification widget.
- Connect the "Data Sampler" widget to the classification widget.
- Double-click the classification widget to configure the algorithm's parameters if needed.
- The model will automatically train using the provided data.

- Drag and drop the "Test andScore" widget into the workspace. C
- onnect the classification widget and the test data from the "Data Sampler" to the "Test andScore" widget.
- The "Test and Score" widget will evaluate the model using the test data and provide performance metrics such as accuracy, precision, recall, and F1 score.

The following tables summarize the results obtained after prediction and classification

TABLE 3.26: Model Evaluation Metrics Dataset1

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC
Tree	8.903	0.018	0.915	0.932	0.932	0.932	0.932	0.896
Neural Network	12.835	0.220	1.013	0.942	0.934	0.945	0.942	0.911
kNN	1.365	39.605	0.992	0.981	0.981	0.981	0.981	0.971
Naive Bayes	1.055	0.153	0.687	0.542	0.675	0.907	0.542	0.495
Gradient Boosting	1135.577	2.012	0.997	0.985	0.985	0.985	0.985	0.977
Random Forest	14.270	0.475	0.993	0.978	0.978	0.978	0.978	0.966
AdaBoost	11.618	0.187	0.881	0.958	0.958	0.958	0.958	0.935

TABLE 3.27: Model Evaluation Metrics Dataset2

Model	Train	Test	AUC	CA	F1	Precision	Recall	MCC
Tree	5.348	0.015	3.021	0.957	0.954	0.955	0.957	0.942
Neural Network	9.917	0.166	3.063	0.932	0.926	0.923	0.932	0.909
kNN	0.413	8.353	3.054	0.950	0.950	0.950	0.950	0.933
Naive Bayes	1.177	0.181	3.050	0.931	0.932	0.936	0.931	0.908
Gradient Boosting	987.593	1.313	3.074	0.994	0.994	0.994	0.994	0.991
Random Forest	13.280	0.369	3.072	0.986	0.986	0.986	0.986	0.981
AdaBoost	11.854	0.130	3.030	0.974	0.974	0.974	0.974	0.965

Where

- **Train:**Train time (s), **Test:**Test time (s), **AUC:** Area under ROC curve, **CA:** Classification accuracy , **Prec:**Precision ,**MCC:** Matthews correlation coefficient

3.4.5 Data Mining Using the Knime Tool

1-Preprocessing

For dataset1

- Open KNIME.
- Drag and drop the "Csv Reader" node from the Node Repository into the workspace.
- Double-click on the "Csv Reader" node to configure it.
- Click "Browse" to load your dataset Click "Apply" and then "OK".
- Drag and drop the "Row filter" node into the workspace.
- Connect the "Csv Reader" node to the "Row filter" node.
- Double-click the "Row filter" node to configure how to handle missing values (remove rows with unknown or notexistes values). for each column contains missing values we use a "Row filter" Click "Apply" and then "OK".Drag and drop the "One to Many" node into the workspace (this node converts categorical varIFbles into binary columns).
- Drag and drop the "Category to number" node into the workspace (this node converts categorical varIFbles into integer columns).
- Double-click the "Category to number" node to select the columns you want to convert from nominal to numeric.
- Click "Apply" and then "OK"
- Drag and drop the "Normalizer" node into the workspace.

- Connect the "category to number" node to the "Normalizer" node.
- Double-click the "Normalizer" node to configure it.
- Select the "Centralizer" option (which centers the data by subtracting the mean).
- Click "Apply" and then "OK".
- Drag and drop the "PCA" node into the workspace.
- Connect the "Normalizer" node to the "PCA" node.
- Double-click the "PCA" node to configure it.
- Click "Apply" and then "OK".

For dataset2

We feeling the same steps but we use only "csv reader", "centralizer" and "CPA" nodes

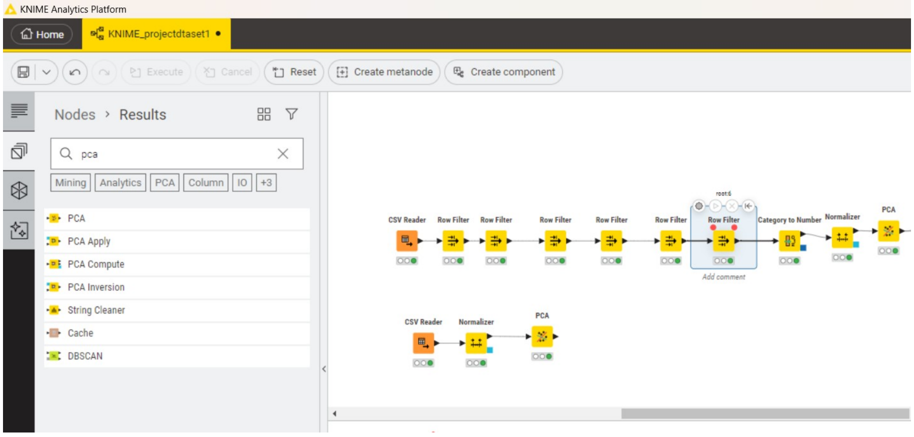


FIGURE 3.12: preprocessing datasets

At the end of the preprocessing on the two datasets, we get this result

TABLE 3.28: Preprocessing Result

Dataset	II	FI	IF	Type	FaPCA	M V
Dataset1	41118	27027	21	NOM+NUM	12	14091
Dataset2	30000	30000	25	NOM+NUM	10	0

2-Clustering dataset To cluster datasets in Knime, We will use the results of preprocessing and use the Two algorithms follow the next steps

- **Choose Clustering Algorithms** In the Node Repository, find the nodes for k-means and k-medoids algorithms.
- **Configure the Algorithms** Drag the k-means node and k-medoids node to the workflow area. Double-click on each node to configure its parameters. For k-means, we need to specify the number of clusters (k), while for k-medoids, we must use "distance matrix calculate" to choose a distance metric.

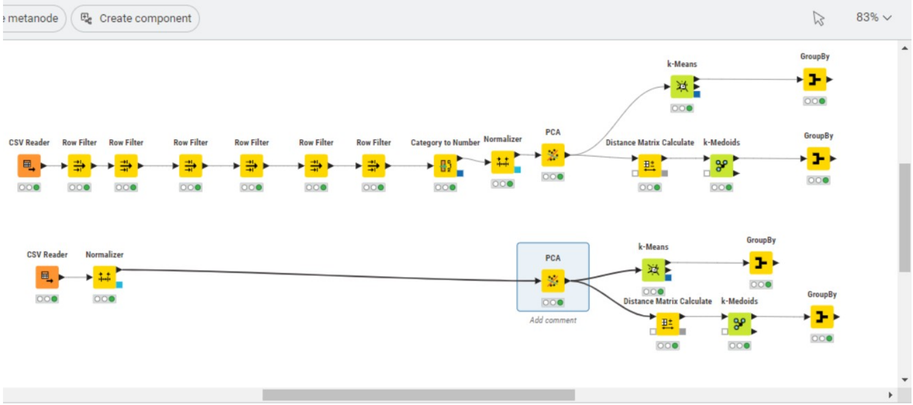


FIGURE 3.13: Clustering datasets

After the clustering process is complete, We should be able to view the results

TABLE 3.29: Clustering Results of Dataset1

Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
K-Means	7864 items	8020 items	4683 items	2768 items	3692 items
k-medoids	4031 items	10385 items	7568 items	4119 items	924 items

TABLE 3.30: Clustering Results of Dataset2

Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
K-Means	3763 items	2873 items	7624 items	6725 items	9015 items
k-medoids	1188 items	3166 items	11262 items	4500 items	9884 items

3-Classification To classify dataset with Knime tool we use the product of K-Means clustering and feel this steps:

- **Partitioning** Use Partitioning nodes such as Partitioning or Stratified Partitioning to split your dataset into training and testing sets.
- **Decision Tree Learning** drag and drop the Decision Tree Learner node,scorer and decision tree predictor to train a decision tree model. Configure this node by selecting the approprIFte target column and any other relevant settings.
- repeat this step for all algorithm uses in classification process the next figure shows clearly .

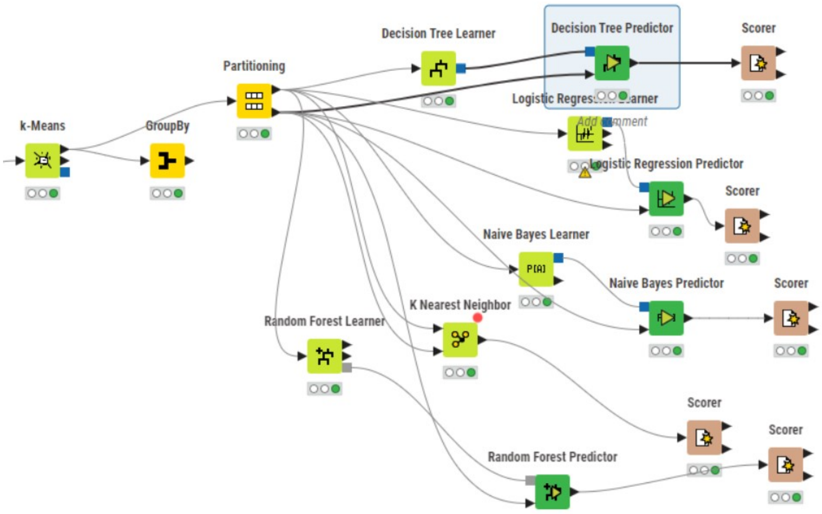


FIGURE 3.14: Classification with Knime tool

. Model Evaluation: Assess the performance of your model using nodes such as Scorer or recall , precision overall...etc . These nodes will provide various metrics to evaluate the performance of your decision tree model.

The following tables summarize the results obtained after prediction and classification: **Dataset1**

TABLE 3.31: Performance Metrics Decision Tree Learner

Cluster	**TP**	**FP**	**TN**	**FN**	**R**	**P**	**S**	**Sp**	**F**
cluster_0	1568	3	3832	3	0.998	0.998	0.998	0.999	0.998
cluster_1	1619	1	3786	0	1.000	0.999	1.000	1.000	1.000
cluster_2	923	0	4483	0	1.000	1.000	1.000	1.000	1.000
cluster_3	565	2	4836	3	0.995	0.996	0.995	1.000	0.996
cluster_4	725	0	4681	0	1.000	1.000	1.000	1.000	1.000

TABLE 3.32: Performance Metrics Logistics Regression Learner

Cluster	**TP**	**FP**	**TN**	**FN**	**R**	**P**	**S**	**Sp**	**F**
cluster_0	1569	0	3835	2	0.998	1.0	0.999	1.0	0.999
cluster_1	1619	1	3786	0	1.0	0.999	1.0	0.999	0.999
cluster_2	923	0	4483	0	1.0	1.0	1.0	1.0	1.0
cluster_3	568	1	4837	0	1.0	0.998	1.0	0.999	0.999
cluster_4	725	0	4681	0	1.0	1.0	1.0	1.0	1.0

TABLE 3.33: Performance Metrics Naive Bayes Learner

Cluster	**TP**	**FP**	**TN**	**FN**	**R**	**P**	**S**	**Sp**	**F**
cluster_0	1485	5	3830	86	0.945	0.996	0.945	0.998	0.970
cluster_1	1617	0	3787	2	0.998	1.0	0.998	1.0	0.999
cluster_2	923	56	4427	0	1.0	0.942	1.0	0.987	0.970
cluster_3	513	85	4753	55	0.903	0.857	0.903	0.982	0.879
cluster_4	722	0	4681	3	0.995	1.0	0.995	1.0	0.997

TABLE 3.34: Performance Metrics K Nearst Nieghbor

Cluster	**TP**	**FP**	**TN**	**FN**	**R**	**P**	**S**	**Sp**	**F**
cluster_0	1563	2	3833	8	0.994	0.998	0.994	0.999	0.996
cluster_1	1619	1	3786	0	1.0	0.999	1.0	0.999	0.999
cluster_2	923	0	4483	0	1.0	1.0	1.0	1.0	1.0
cluster_3	566	8	4830	2	0.996	0.986	0.996	0.998	0.991
cluster_4	724	0	4681	1	0.998	1.0	0.998	1.0	0.999

TABLE 3.35: Performance Metrics Random Forest learner

Cluster	**TP**	**FP**	**TN**	**FN**	**R**	**P**	**S**	**Sp**	**F**
cluster_0	1570	4	3831	1	0.999	0.997	0.999	0.999	0.998
cluster_1	1619	1	3786	0	1.0	0.999	1.0	0.999	0.999
cluster_2	923	0	4483	0	1.0	1.0	1.0	1.0	1.0
cluster_3	564	0	4838	4	0.992	1.0	0.992	1.0	0.996
cluster_4	725	0	4681	0	1.0	1.0	1.0	1.0	1.0

Dataset2

TABLE 3.36: Performance Metrics Decision Tree Learner

Cluster	TP	FP	TN	FN	R	P	S	Sp	F
cluster_0	746	0	5254	0	1.0	1.0	1.0	1.0	1.0
cluster_1	532	0	5468	0	1.0	1.0	1.0	1.0	1.0
cluster_2	1462	8	4525	5	0.996	0.994	0.996	0.998	0.995
cluster_3	1349	5	4638	8	0.994	0.996	0.994	0.998	0.995
cluster_4	1898	0	4102	0	1.0	1.0	1.0	1.0	1.0

TABLE 3.37: Performance Metrics Logistics Regression Learner

Cluster	TP	FP	TN	FN	R	P	S	Sp	F
cluster_0	746	0	5254	0	1.0	1.0	1.0	1.0	1.0
cluster_1	532	0	5468	0	1.0	1.0	1.0	1.0	1.0
cluster_2	1467	8	4525	0	1.0	0.994	1.0	0.998	0.997
cluster_3	1349	0	4643	8	0.994	1.0	0.994	1.0	0.997
cluster_4	1898	0	4102	0	1.0	1.0	1.0	1.0	1.0

TABLE 3.38: Performance Metrics aive Bayes Learner

Cluster	TP	FP	TN	FN	R	P	S	Sp	F
cluster_0	746	6	5248	0	1.0	0.992	1.0	0.998	0.995
cluster_1	532	2	5466	0	1.0	0.996	1.0	0.999	0.998
cluster_2	1461	16	4517	6	0.995	0.989	0.995	0.996	0.992
cluster_3	1340	1	4642	17	0.987	0.999	0.987	0.999	0.993
cluster_4	1896	0	4102	2	0.998	1.0	0.998	1.0	0.999

TABLE 3.39: Performance Metrics K Nearst Nieghbor

Cluster	TP	FP	TN	FN	R	P	S	Sp	F
cluster_0	746	0	5254	0	1.0	1.0	1.0	1.0	1.0
cluster_1	532	0	5468	0	1.0	1.0	1.0	1.0	1.0
cluster_2	1464	12	4521	3	0.997	0.991	0.997	0.997	0.994
cluster_3	1345	3	4640	12	0.991	0.997	0.991	0.999	0.994
cluster_4	1898	0	4102	0	1.0	1.0	1.0	1.0	1.0

TABLE 3.40: Performance Metrics

Cluster	TP	FP	TN	FN	R	P	S	Sp	F
cluster_0	746	0	5254	0	1.0	1.0	1.0	1.0	1.0
cluster_1	532	0	5468	0	1.0	1.0	1.0	1.0	1.0
cluster_2	1466	6	4527	1	0.999	0.99	0.999	0.998	0.997
cluster_3	1351	1	4642	6	0.995	0.999	0.995	0.999	0.997
cluster_4	1898	0	4102	0	1.0	1.0	1.0	1.0	1.0

where TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative, R: Recall, P: Precision, S: Sensitivity, Sp: Specificity, F: F-measure.

3.5 Discussion

We discuss the results between four data mining tools using three important phases: preprocessing, clustering, and classification.

3.5.1 Preprocessing Results Analysis

In this section, we compare the preprocessing results of RapidMiner, Weka, Orange, and KNIME using two datasets. The metrics used for comparison include

Type (Nominal or Numeric), Initial Instances (II), Final instances (FI), Initial features (IF), Features after PCA (FaPCA), and Missing Values (MV).

Table 3.41 depicts the preprocessing results for each tool.

TABLE 3.41: Result of the preprocessing phase of Dataset 1

Tool	RapidMiner	Weka	Orange	KNIME
Type	Nominal + numeric	Nominal + numeric	Nominal + numeric	Nominal + numeric
II	41118	41118	41118	41118
FI	27027	27027	27027	27027
IF	21	21	21	21
FaPCA	16	16	15	12
MV	14091	14091	14091	14091

TABLE 3.42: Result of preprocessing phase for Dataset 2

Tool	RapidMiner	Weka	Orange	KNIME
Type	Nominal + numeric	Nominal + numeric	Nominal + numeric	Nominal + numeric
II	30000	30000	30000	30000
FI	30000	30000	30000	30000
IF	25	25	25	25
FaPCA	23	17	17	10
MV	0	0	0	0

According to the comparison of the preprocessing results, we can deduct the following properties:

RapidMiner

Strengths

. It supports various dimensionality reduction algorithms, such as Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), which are effective in reducing the feature space while preserving the important information in the data.

Weaknesses

- High resource usage might make it less efficient for very large datasets.
- Limited free features could restrict its utility in cost-sensitive environments.
- Slightly less efficient in dimension reduction in two datasets, for example, Dataset2, defines 23 components from 25 features.

Weka

Strengths

- Retained final instances and managed missing values effectively.
- Basic visualization capabilities might give deeper exploratory analysis.

Weaknesses

- Slightly less efficient in feature reduction, for example, Dataset 1, reduces only 5 features from 21 features.

Orange

Strengths

- User-friendly and visually appealing, making it easy to understand and use preprocessing results.

Weaknesses

- Slightly less efficient in feature reduction, for example, Dataset 1, reduces only 5 features from 21 features.
- Limited preprocessing options might require additional external steps for comprehensive data preparation.

KNIME

Strengths

- Most effective in features reduction through PCA, reducing Dataset 1 to 12 featuress and Dataset 2 to 10 featuress.
- Designed to handle larger datasets efficiently.

Weaknesses

- Steeper learning curve and complex GUI might deter new users.
- Resource-intensive, potentIFlly requiring significant computational power for complex workflows

Based on the preprocessing results, it is clear that each tool possesses distinct advantages and disadvantages, for example, we deduct that

Knime effectively manages feature reduction while preserving a significant number of characteristics after performing PCA. This makes it suited for situations when keeping more features is advantageous.

Weka offers a balanced approach by providing efficient preprocessing capabilities, however, it is not as assertive in reducing features.

Orange provides a superior user experience and produces satisfactory preprocessing outcomes, while it may necessitate additional tools for thorough data management.

RapidMiner needs substantial commitment of time and computational resources for learning.

The necessity for enhanced visualisation, user-friendly interfaces, aggressive feature reduction, or handling big datasets are some of the project-specific needs that can inform the choice of tool.

3.5.2 Clustering Results Analysis

In order to assess the clustering skills of RapidMiner, Weka, Orange, and KNIME, we examine the clustering outcomes on two datasets by utilising different clustering algorithms offered by each software.

Regarding the labelled dataset 1, each tool gives proper clusters, which are different to the original one in the number of instances.

The clustering techniques utilised encompass K-Means, K-Medoids, DBSCAN, Hierarchical Clustering, and more algorithms tailored to each unique tool. The evaluation focuses on the allocation of instances among clusters, the efficacy of the algorithms, and their capacity to successfully manage the data.

TABLE 3.43: Clustering Results for **Dataset 1**

Tool	Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
RapidMiner	K-means-fast	17514 items	2 items	1191 items	1041 items	7279 items
	K-Means	7279 items	2 items	1191 items	1039 items	17516 items
	X-Means	17514 items	2 items	1191 items	1041 items	7279 items
	K-Means (H2O)	4357 items	2 items	4045 items	15662 items	2961 items
	K-Medoids	135 items	6076 items	17139 items	2614 items	1063 items
	Random Clustering	5399 items	5470 items	5387 items	5465 items	5306 items

Tool	Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Weka	EM	6557 items	3572 items	1806 items	12423 items	2669 items
	Canopy	12600 items	4838 items	5900 items	2512 items	1177 items
	FarthestFirst	21361 items	588 items	557 items	2565 items	1956 items
	Filtered-Clusterer	44178 items	8866 items	1187 items	9801 items	2985 items
	Make density-based	4186 items	8417 items	1191 items	10247 items	9286 items
	Simple-KMeans	4178 items	8866 items	1187 items	9801 items	2995 items
Orange	Hierarchical	2 items	1190 items	3494 items	8077 items	14264 items
	K-Means	4030 items	10169 items	1191 items	11622 items	2 items
	DBSCAN	25685 items	1048 items	4 items	3 items	6 items
	Interactive K-Means	9598 items	12153 items	4085 items	171 items	1020 items
KNIME	K-Means	7864 items	8020 items	4683 items	2768 items	3692 items
	K-Medoids	4031 items	10385 items	7568 items	4119 items	924 items

TABLE 3.44: Clustering Results for **Dataset 2**

Tool	Algorithm	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
RapidMiner	K-means-fast	25346 items	1 item	1 item	4650 items	2 items
	K-Means	15728 items	3 items	10606 items	3661 items	2 items
	X-Means	25346 items	1 item	1 item	4650 items	2 items
	K-Means (H2O)	10057 items	4 items	1164 items	15646 items	3129 items
	K-Medoids	2474 items	4172 items	479 items	22348 items	527 items
	Random Clustering	5972 items	6076 items	5984 items	6080 items	5888 items
Weka	EM	1526 items	366 items	21747 items	4822 items	1539 items
	Canopy	16968 items	6636 items	18 items	27 items	6351 items
	FarthestFirst	23359 items	1 item	3 items	1 item	6636 items
	Filtered-Clusterer	9530 items	7065 items	2449 items	4187 items	6769 items
	Make density-based	9005 items	6829 items	2574 items	4347 items	7245 items
	Simple-KMeans	9530 items	7065 items	2449 items	4187 items	6769 items
Orange	Hierarchical	593 items	9224 items	98 items	82 items	20003 items
	K-Means	6525 items	8811 items	6610 items	7733 items	294 items
	DBSCAN	580 items	6721 items	12147 items	7817 items	2735 items
	Interactive K-Means	29018 items	3 items	6 items	6 items	4 items
KNIME	K-Means	3763 items	2873 items	7624 items	6725 items	9015 items
	K-Medoids	1188 items	3166 items	11262 items	4500 items	9884 items

RapidMiner

Strengths

- Variety of Algorithms: RapidMiner offers multiple clustering algorithms such as K-Means, X-Means, K-Means (H2O), and K-Medoids.
- Consistent Clustering: The K-means-fast and X-Means algorithms provided consistent results, indicating stable performance.
- K-Means (H2O): This algorithm effectively distributed instances across clusters, showing its capability in handling complex datasets

Weaknesses

- Small Clusters: Some algorithms resulted in very small clusters (2 items), which may indicate sensitivity to initial conditions or the presence of outliers.
- Random Clustering: As expected, produced evenly distributed clusters without meaningful segmentation, useful primarily for benchmarking.

Weka

Strengths

- Algorithm Diversity: Weka provides a range of clustering algorithms, including EM, Canopy, FarthestFirst, and density-based clustering, offering flexibility.
- Efficient Clustering: Algorithms like FarthestFirst quickly grouped most instances into large clusters, demonstrating efficiency.
- Density-Based Clustering: The density-based clustering algorithm effectively identified dense regions in the data.

Weaknesses

- Cluster Variability: Some algorithms showed significant variability in cluster sizes (one instance in a cluster), indicating sensitivity to parameter settings.

. Inconsistent Results: Different algorithms produced widely varying results, highlighting the importance of careful parameter tuning.

Orange

Strengths

- User-Friendly Interface: Orange provides an intuitive and interactive interface, making clustering results easy to understand and visualize.
- Hierarchical Clustering: This method offered detailed insights into the data structure, useful for exploratory analysis
- Interactive K-Means: Allowed for user-guided adjustments, enhancing cluster interpretability and customization.

Weaknesses

- Sparse Clusters: Some algorithms produced clusters with very few instances, suggesting overfitting or inadequate parameter settings.
- Mixed Performance: While DBSCAN handled noise well, other algorithms like Interactive K-Means showed inconsistent clustering performance

KNIME

Strengths

- Balanced Clustering: KNIME's K-Means and K-Medoids algorithms provided balanced cluster distributions, effectively grouping instances.
- Scalability: KNIME handled large datasets efficiently, maintaining performance and cluster quality.

Weaknesses

- Moderate Complexity: KNIME's clustering algorithms, while effective, showed moderate complexity in setup and parameter tuning. Cluster Variability: Similar to other tools, KNIME also exhibited some variability in cluster sizes across different algorithms.

Every tool shows both strengths and drawbacks in terms of clustering performance. RapidMiner and KNIME demonstrated consistent and equitable clustering skills, particularly when utilising methods such as K-Means (H2O) and K-Means. Weka offered a wide array of techniques but showed significant variation in the sizes of clusters. Orange provided intuitive and engaging functionalities, despite certain algorithms yielding sparse clusters. The selection of the tool and technique is contingent upon the specific needs of the dataset and the intended clustering results.

3.5.3 Classification Results Analysis

Next table presents a comparative analysis of various data mining platforms RapidMiner, Weka, Orange, and KNIME—by evaluating their performance using different models and metrics. The main focus is on identifying the top-performing models and understanding their efficiency through various performance metrics.

TABLE 3.45: Performance Metrics and Tools for Data Mining Platforms

Metric/Tool	**RapidMiner**	**Weka**	**Orange**	**KNIME**
Top Performing Models	Gradient Boosted Trees, Random Forest	Random Forest, Naive Bayes	Gradient Boosting, kNN	Decision Tree Learner, Logistic Regression
Gradient Boosted Trees	CE: 13.2% ±0.3%	Not Evaluated	CA: 0.985, AUC: 0.997, F1: 0.985	Not Evaluated
Random Forest	CE: 19.9% ±0.7%	TP Rate: 0.991, FP Rate: 0.002, Precision: 0.991	CA: 0.978, AUC: 0.993, F1: 0.978	Accuracy: 1.000, Precision: 1.000, F1: 1.000
Naive Bayes	CE: 31.80% ±0.60%	TP Rate: 0.927, FP Rate: 0.012, Precision: 0.934	CA: 0.542, AUC: 0.687, F1: 0.675	Accuracy: 0.998, Precision: 0.998, F1: 0.998
Logistic Regression	CE: 44.16% ±0.80%	Not Evaluated	Not Evaluated	Accuracy: 1.000, Precision: 0.999, F1: 1.000
Decision Tree	CE: 33.66% ±0.69%	Not Evaluated	CA: 0.932, AUC: 0.915, F1: 0.932	Accuracy: 0.998, Precision: 0.998, F1: 0.998
kNN	Not Evaluated	Not Evaluated	CA: 0.981, AUC: 0.992, F1: 0.981	Not Evaluated
Deep Learning	CE: 20.78% ±0.23%	Not Evaluated	CA: 0.932, AUC: 1.013, F1: 0.934	Not Evaluated
Training Time	Varies by model (e.g., 3s for Naive Bayes, 2m32s for Gradient Boosted Trees)	Generally quick (e.g., 0.09s for Naive Bayes, 8.94s for Random Forest)	Varies by model (e.g., 1.365s for kNN, 1135.577s for Gradient Boosting)	Varies by model

Metric/Tool	**RapidMiner**	**Weka**	**Orange**	**KNIME**
Scoring Time	Varies by model (e.g., 203ms for Naive Bayes, 770ms for Gradient Boosted Trees)	Generally quick (e.g., 20ms for Decision Tree, 0.02s for ZeroR)	Varies by model (e.g., 39.605 for kNN, 2.012 for Gradient Boosting)	Varies by model
Model Building Time	Varies by model	Generally quick	Varies by model	Varies by model
Overall Accuracy	High for Gradient Boosted Trees, Random Forest	Very High for Random Forest, Naive Bayes	High for Gradient Boosting, kNN	Very High for Decision Tree Learner, Logistic Regression

*where***CE (Classification Error) and CA (Correctly Classified Instances)**

To provide a detailed comparison of the performance of classification algorithms across RapidMiner, WEKA, Orange, and KNIME, let's focus on the key metrics: accuracy, speed, memory usage, algorithm variety, and usability.

TABLE 3.46: Detailed Performance Comparison of Classification Algorithms

Feature	**RapidMiner**	**WEKA**	**Orange**	**KNIME**
Accuracy	High	High	Moderate	High
Speed	High	Moderate	High	High
Memory Usage	Moderate	High	Low	Moderate
Algorithm Variety	Extensive	Extensive	Moderate	Extensive
Usability	High	Moderate	High	High
Integration	High	Moderate	Moderate	High
Community Support	High	High	Moderate	High
Documentation	Extensive	Extensive	Good	Extensive

Analysis

Accuracy

- *RapidMiner:* is renowned for its excellent accuracy because of its sophisticated optimisation methods and algorithms. Ideal for tasks needing exact outcomes.
- *WEKA:* This powerful collection of algorithms, which is especially useful for scholarly and research endeavours, also provides excellent accuracy.
- *Orange:* Offers a moderate level of precision. Although it is easy to use, it might not be as accurate as WEKA or RapidMiner.
- *KNIME:* Uses a variety of algorithms and data processing power to deliver excellent accuracy.

Speed

- *KNIME and RapidMiner:* Both are designed for high-speed processing, which makes them appropriate for real-time analytics and big datasets.
- *WEKA:* Moderate speed; huge datasets or complicated models may cause it to lag.
- *Orange:* Quick speed, particularly preferred in contexts involving interactive and visual data analysis.

Memory usage

- *Orange:* is characterised by its low memory utilisation, which makes it highly efficient for systems that have limited resources.
- Both*RapidMiner and KNIME* have a moderate memory utilisation, effectively balancing performance and resource consumption.
- *WEKA:* Utilises a significant amount of memory, potentially affecting the efficiency of operation on less capable devices.

Algorithm Variety •

- *RapidMiner:* offers a wide range of classification algorithms, such as decision trees, logistic regression, and support vector machines.
- WEKA: provides a wide range of algorithms, including J48 (C4.5 decision tree), Naive Bayes, and Random Forest.
- *Orange:* Moderate diversity, covering fundamental algorithms like k-nearest neighbors, decision trees, and Naive Bayes.
- *KNIME:* offers a wide range of options, such as decision trees, neural networks, and logistic regression.

Ease of Use

- *RapidMiner*, *Orange* and *KNIME:* offer high usability because to their intuitive interfaces and streamlined operations.
- *WEKA:* Moderate usability with a steeper learning curve, more appropriate for people with a technical background.

Integration

- *RapidMiner* and *KNIME* have strong integration capabilities with a wide range of databases, big data tools, and other applications.
- *Weka* and *Orange*have a moderate level of integration, which makes them appropriate for ordinary applications. However, for more complicated integrations, further setup may be needed.

Documentation

- *RapidMiner*, *WEKA*, and *KNIME* is available and includes comprehensive resources like as tutorials, user guides, and technical manuals
- *Orange:* The documentation is good, but not as thorough as the others.

.

Classification Algorithms Overview

***RapidMiner*:** provides a diverse selection of classification techniques, such as decision trees, logistic regression, and support vector machines. The combination of its except-ional precision and rapidity, together with its user-friendly interface, makes it a preferred option for several data scientists and analysts.

***WEKA*:** is renowned for its vast array of machine learning methods, including several categorization techniques such as J48 (C4.5 decision tree), Naive Bayes, and Random Forest. Although it provides excellent accuracy, the limited user interface and slower performance speed may provide challenges for some users.

***Orange*:** is a software that offers many classification techniques, including k-nearest neighbours, decision trees, and Naive Bayes. It is designed to prioritise visual programming and user-friendliness. The performance of the system is incredibly efficient, while its accuracy may be somewhat inferior when compared to RapidMiner and WEKA.

***KNIME*:** incorporates a diverse range of classification techniques, such as decision trees, neural networks, and logistic regression. It offers a blend of exceptional precision and rapidity, coupled with a user-friendly graphical interface, making it appropriate for users of all levels of expertise.

3.6 Conclusion

An analysis of the tools RapidMiner, Weka, Orange, and KNIME—uncovered clear advantages and disadvantages. RapidMiner demonstrated strong preprocessing capabilities and user-friendly workflow management, making it well-suited for handling huge datasets. Weka, known for its vast array of machine learning algorithms, offers in-depth analysis and exceptional precision in classification tasks, particularly with its RandomForest and NaiveBayes classifiers. The user-friendly interface and great visualisations of Orange helped a better comprehension of clustering findings. KNIME's modular architecture facilitated the smooth incorporation of many analytical techniques, hence improving its adaptability and expandability.

In general, the experimental findings validated that no one tool or algorithm is generally superior. Instead, the selection of the appropriate tool or algorithm relies on the unique demands of the given activity. The thorough assessment and comparison presented in this chapter may be used as a reference for choosing suitable data mining tools and techniques for future study and practical applications.

General Conclusion

Our study compares the data mining tools RapidMiner, Weka, Orange, and KNIME, emphasising their varied capabilities and applications. Every tool possesses distinct advantages and disadvantages that are tailored to specific data mining jobs. RapidMiner is notable for its extensive range of features and user-friendly interface, making it ideal for those in need of a powerful, all-encompassing solution. Weka is renowned for its vast array of machine-learning algorithms, making it highly suitable for academic and scientific endeavours. Orange's visual programming and intuitive interface make it easily accessible to beginners and individuals who prioritise user-friendliness and the ability to see data. KNIME is highly scalable and offers excellent integration capabilities, making it an ideal solution for professional settings that require intricate processes and extensive data processing.

The results of dynamic analysis demonstrate that although all tools proficiently manage data preparation, clustering, and classification tasks, their effectiveness is contingent upon the particular datasets and algorithms employed. RapidMiner and KNIME exhibit exceptional proficiency in managing extensive datasets and intricate algorithms, while Weka and Orange offer important insights through their comprehensive analysis capabilities and visualisation tools.

Ultimately, the selection of a data mining tool should be based on the particular demands of the task, the user's proficiency, and the computational resources at their disposal.

Our comparative study between four common free data mining software tools allows organisations and researchers to choose the most suitable tool for their requirements.

This Master's thesis offers a thorough basis for comprehending and enhancing the process of software requirements engineering. The future work emphasises the potential for substantial progress in this subject, propelled by technology breakthroughs, worldwide cooperation, and a dedication to ethical principles. We will extend our study by applying different types of data mining such as, multimedia mining, text mining, web mining, and graph mining.

By considering these prospects, researchers and professionals can further improve the calibre, effectiveness, and influence of software requirements engineering, ultimately resulting in superior software products and increased stakeholder satisfaction.

Bibliography

[1] Venkateswarlu Pynam, R Roje Spanadna **and** Kolli Srikanth. ?An extensive study of data analysis tools (rapid miner, weka, r tool, knime, orange)? **in***Int. J. Comput. Sci. Eng*: 5.9 (2018), **pages** 4–11.

[2] Toshiaki Aizawa. ?Decomposition of Improvements in Infant Mortality in Asian Developing Countries Over Three Decades? **in***Demography*: 58.1 (2021), **pages** 137–163. DOI: 10.1215/00703370-8931544.

[3] Arun K Pujari. *Data mining techniques*. Universities press, 2001.

[4] Jafar Alzubi, Anand Nayyar **and** Akshi Kumar. ?Machine learning from theory to algorithms: an overview? **in***Journal of physics: conference series*: **volume** 1142. IOP Publishing. 2018, **page** 012012.

[5] Aggoune Aicha. ?Data mining? **in***Course of 2nd years Computer science*: University of Guelma (2024), **page** 4.

[6] Hadley Wickham **and** Hadley Wickham. *Data analysis*. Springer, 2016.

[7] Yi Lu **andothers**. ?Aria: a fast and practical deterministic OLTP database? **in**(2020).

[8] Imad Abugessaisa **and** Takeya Kasukawa. *Practical guide to life science databases*. Springer, 2021.

[9] Colin Shearer. ?The CRISP-DM model: The new blueprint for data mining? **in***Journal of Data Warehousing*: 5.4 (2000), **pages** 13–22.

[10] Rob Petersen. *6 essential steps to the data mining process*. https://barnraisersllc.com/2018/10/01/data-mining-process-essential-steps/. Accessed: June 10th, 2024.

[11] SMART VISION EUROPE. *What is the CRISP-DM methodology?* https://www.sv-europe.com/crisp-dm-methodology/. Accessed: June 10th, 2024.

[12] www.proglobalbusinesssolutions.com. *Six Steps in CRISP DM - The Standard Data Mining Process*. https://www.proglobalbusinesssolutions.com/six-steps-in-crisp-dm-the-standard-data-mining-process/. Accessed: June 10th, 2024.

[13] Yong Zhong **andothers**. ?A systematic survey of data mining and big data analysis in internet of things? **in***The Journal of Supercomputing*: 78.17 (2022), **pages** 18405–1845

[14] Dr. M. Venkateswara Rao Mylavarapu Kalyan Ram **and** Challapalli Sujana. ?An Overview on Multimedia Data Mining and Its Relevance Today? **in***International Journal of Computer Science Trends and Technology (IJCST)*: 5.3 (2017), **pages** 108–109. ISSN: 2347-8578. URL: http://www.ijcstjournal.org.

[15] S. Kalaichelvi. ?Web Mining Classification: a Survey? **in***International Journal of Engineering Research and Technology (IJERT)*: 3.10 (2014), **page** 1025. ISSN: 2278-0181. URL: https://www.ijert.org.

[16] thabitha. *Data Mining Graphs and Networks*. https://www.geeksforgeeks.org/data-mining-graphs-and-networks/. Accessed: 04/28/2024.

[17] javatpoint. *Types of Data Mining*. https://www.javatpoint.com/types-of-data-mining. Accessed: June 10th, 2024.

[18] WILL HILLIER. *What Is Descriptive Analytics? A Complete Guide*. https://careerfoundry.com/en/blog/data-analytics/descriptive-analytics/. Accessed: June 10th, 2024.

[19] Catherine Cote. *WHAT IS PREDICTIVE ANALYTICS? 5 EXAMPLES*. https://online.hbs.edu/blog/post/predictive-analytics. Accessed: June 10th, 2024.

[20] Ritinder Kaur. *Descriptive vs Predictive vs Prescriptive vs Diagnostic Analytics*. https://www.selecthub.com/business-intelligence/predictive-descriptive-prescriptive-analytics/. Accessed: June 10th, 2024.

[21] Utkarsh. *Classification in Data Mining*. https://www.scaler.com/topics/data-mining-tutorial/classification-in-data-mining/. Accessed: June 10th, 2024.

[22] sagar shukla. *Regression in machine learning*. https://www.geeksforgeeks.org/regression-in-machine-learning/. Accessed: June 10th, 2024.

[23] Eoghan Keany. *The Ultimate Guide for Clustering Mixed Data*. TheUltimateGuideforClusteri Accessed: June 10th, 2024.

[24] Nirajan Khadka. *THE ULTIMATE GUIDE TO ASSOCIATION RULE ANALYSIS*. https://dataaspirant.com/association-rule-analysis/. Accessed: June 10th, 2024.

[25] Ofem Eteng. *Data Summarization in Data Mining Simplified 101*. DataSummarizationinDataMi Accessed: June 10th, 2024.

[26] Gaudenz Boesch. *What is Pattern Recognition? A Gentle Introduction (2024)*. https://viso.ai/deep-learning/pattern-recognition. Accessed: June 10th, 2024.

[27] Alicia Horsch. *Hypothesis testing for data scientists*. https://towardsdatascience.com/hypothesis-testing-for-data-scientists-everything-you-need-to-know-8c36ddde4cd2. Accessed: June 10th, 2024.

[28] IndeedEditorial Team. *6 Statistical Methods (Plus Definition and Importance)*. https://www.indeed.com/career-advice/career-development/statistical-methods. Accessed: June 10th, 2024.

[29] Pritha Bhandari. *Descriptive Statistics | Definitions, Types, Examples*. https://www.scribbr.com/statistics/descriptive-statistics/. Accessed: June 10th, 2024.

[30] Jim Frost. *Mean, Median, and Mode: Measures of Central Tendency*. https://statisticsbyjim.com/basics/measures-central-tendency-mean-median-mode/. Accessed: June 10th, 2024.

[31] Calculator.net. *Standard Deviation Calculator*. StandardDeviationCalculator. Accessed: June 10th, 2024.

[32] Muhammad Hassan. *Regression Analysis– Methods, Types and Examples*. https://researchmethod.net/regression-analysis/. Accessed: June 10th, 2024.

[33] Muhammad Hassan. *Regression Analysis–Methods, Types and Examples*. https://researchmethod.net/inferential-statistics/. Accessed: June 10th, 2024.

[34] Coursera Staff. *What Is Machine Learning? Definition, Types, and Examples*. https://www.coursera.org/articles/what-is-machine-learning/. Accessed: June 5th, 2024.

[35] Pádraig Cunningham, Matthieu Cord **and** Sarah Jane Delany. ?Supervised Learning? **in***Machine Learning Techniques for Multimedia: Case Studies on Organisation and Retrieval*: (2008), **pages** 21–49.

[36] Akash Dubeys. *The Mathematics Behind Principal Component Analysis*. https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643. Accessed: June 10th, 2024.

[37] Kurtis Pykes. *Introduction to Unsupervised Learning*. https://www.datacamp.com/blog/introduction-to-unsupervised-learning. Accessed: June 10th, 2024.

[38] Deval Shah, Abhishek Jha. *Self-Supervised Learning and Its Applications*. https://neptune.ai/blog/self-supervised-learning. Accessed: June 10th, 2024.

[39] Jérémy Robert. *Transfer Learning : Qu'est-ce que c'est ?s*. https://datascientest.com/transfer-learning. Accessed: June 10th, 2024.

[40] IBM. *What is deep learning?*. https://www.ibm.com/topics/deep-learning/. Accessed: June 15th, 2024.

[41] Yash V. Bagal **andothers**. ?Data Mining in Agriculture: A Novel Approach? **in***International Journal of Engineering Research and Technology (IJERT)*: 9.08 (2020), **page** 213. ISSN: 2278-0181. URL: https://www.ijert.org/research/data-mining-in-agriculture-a-novel-approach-IJERTV9IS080107.pdf.

[42] Fernando Lezama. ?Data Mining and Analysis in Power and Energy Systems: An Introduction to Algorithms and Applications? **in***Intelligent Data Mining and Analysis in Power and Energy Systems: Models and Applications for Smarter Efficient Power Systems*: (2023), **pages** 25–44.

[43] Samuel Greengard. *RapidMiner: Product Overview and Insight*. https://www.datamation.com/artificial-intelligence/rapidminer-product-overview-and-insight/. Accessed: June 10th, 2024.

[44] docs.rapidminer.com. *Downloading and installing Weka*. https://docs.rapidminer.com/9.10/studio/installation/index.html. Accessed: June 15th, 2024.

[45] ALTairone. *Create an Altair One Account*. https://admin.altairone.com/register. Accessed: May 24th, 2024.

[46] ALTairone. *RAPIDMINER DOCUMENTATION*. https://docs.rapidminer.com/9.10/studio/getting-started/design-view.html. Accessed: May 30th, 2024.

[47] Weka 3. *Weka 3: Machine Learning Software in Java*. www.cs.waikato.ac.nz/ml/weka/. Accessed: may 26th, 2024.

[48] Ginni. *What is Weka data mining?* https://www.tutorialspoint.com/what-is-weka-data-mining/. Accessed: june 6th, 2024.

[49] weka wiki. *Downloading and installing Weka*. https://waikato.github.io/weka-wiki/downloading_weka/. Accessed: june 3rd, 2024.

[50] xperra.com. *KNIME Analytics Platform*. https://xperra.com/technology/knime.html. Accessed: June 3rd, 2024.

[51] Ginni. *What is Orange Data Mining?* https://www.tutorialspoint.com/what-is-orange-data-mining/. Accessed: June 10th, 2024.

[52] University of Ljubljana. *Suggested Download*. https://orangedatamining.com/download/. Accessed: May 30th, 2024.