Journal of Applied Business and Technology

Analysis of System Requirements and Architecture for Facilitating Table-Based Data Clustering for Non-Technical Users

Yoppy Yunhasnawa[a*], Toga Aldila Cinderatama[a], Candra Bella Vista[a]

[a]Information Technology Department, Politeknik Negeri Malang, Indonesia

Article History
Received
27 August 2023
Received in revised form
2 September 2023
Accepted
3 September 2023
Published Online
30 September 2023

*Corresponding author
yunhasnawa@polinema.ac.id

Abstract
Clustering is one of the key techniques in unsupervised learning analysis, aimed at grouping similar data objects into clusters based on shared characteristics. The broad benefits of clustering are evident across various sectors, such as business, marketing, finance, and many others. However, the complexity of implementing clustering, especially for those without a background in statistics or programming, poses a barrier. The appropriate selection of clustering methods and accurate interpretation of results require a solid understanding of statistics. This research aims to address this issue by crafting a detailed Software Requirements Specification for a user-friendly clustering application, equipped with an intuitive interface and effective tools, based on comprehensive literature study, which finally allowing non-experts to engage in the clustering process without in-depth knowledge of statistics or programming. As such, this study endeavors to provide a practical solution for utilizing clustering without excessive technical impediments.

Keywords: Clustering, Unsupervised Learning, User-friendly Interface, Software Requirements Specification, Data Analysis

DOI: https://doi.org/10.35145/jabt.v4i3.140

1.0 INTRODUCTION

Clustering, as a key technique in unsupervised learning analysis, aims to group data objects with similar characteristics into clusters. This technique provides an insightful view of relationships and hidden patterns within data, which, in turn, can aid in revealing structures that might not be apparent during initial examination (*Garcia-Dias et al. - 2020 - Clustering Analysis.Pdf*, n.d.).

The benefits resulting from the implementation of clustering across various sectors cannot be ignored. In the business domain, applying clustering techniques enables improved customer segmentation based on behavior and preferences (Kansal et al., 2018). With a better understanding of distinct customer groups, tailored marketing strategies can be employed to maximize campaign effectiveness. In the realm of marketing, clustering techniques play a crucial role in developing more focused campaigns and personalizing messages for customer groups most likely to respond positively (Lecturer Fergana State University, Republic Of Uzbekistan, Fergana et al., 2021). In the field of financial analysis, clustering plays a strategic role in identifying patterns in market behavior, aiding in more informed investment decision-making (Linares-Mustarós et al., 2018).

However, while its benefits are evident, it must be acknowledged that the implementation of clustering techniques is not a simple task, especially for individuals with limited background in statistics and programming (Bjälkebring, 2019; Kranzler & Anthony, 2022). The process of selecting appropriate clustering methods based on data characteristics, as well as accurately interpreting clustering results, demands a strong grasp of fundamental statistical concepts. Additionally, key parameters such as the number of clusters also play a crucial role in producing meaningful groups (Trunfio, 2012). Therefore, for non-experts seeking to apply clustering, it becomes critical to recognize the limits of their technical knowledge and to seek approaches that enable them to deeply understand these concepts, or to involve experts when necessary.

In the realm of research and development, utilizing clustering as a tool to reveal patterns and relationships within data carries significant implications. This article encompasses a comprehensive literature review, aiming to enhance comprehension of clustering, delineate its benefits across various common usage contexts, and underscore challenges that individuals lacking technical backgrounds might encounter during application. Moreover, our primary research objective is to furnish a detailed Software Requirements Specification sourced from the literature review process. This specification offers an overview for the implementation of a user-friendly clustering application tailored for non-technical users. Anticipated to surmount technical barriers, this application will furnish an intuitive interface and streamlined workflow, thus empowering non-experts to participate in the clustering process without necessitating an in-depth grasp of statistics or programming. Consequently, this study aims to provide a pragmatic solution for those seeking to capitalize on clustering's advantages without excessive technical impediments.

2.0 LITERATURE REVIEW

Clustering Algorithm

Clustering is a technique in machine learning and data analysis used to group similar data objects based on their inherent characteristics or properties. Clustering is an unsupervised learning method, which means it doesn't rely on labeled data to make predictions or groupings. To generate clusters from the available data, specific clustering algorithms are required. These clustering algorithms aim to identify patterns or structures within the data by dividing it into distinct groups known as clusters (Rodriguez et al., 2019). Each cluster typically consists of data objects that are more similar to each other compared to objects in other clusters. The goal is to maximize intra-cluster similarity and minimize inter-cluster similarity (Ahmed et al., 2020).

There are various clustering algorithms available, each with its own strengths and weaknesses. Some popular algorithms include:

1. **K-means**: This algorithm aims to divide data into a predetermined number of clusters, where each data object is assigned to the cluster with the nearest mean value (Sinaga & Yang, 2020).
2. **Hierarchical clustering**: This algorithm constructs a hierarchy of clusters by merging or splitting clusters based on their similarities. This can result in a tree-like structure called a dendrogram (Cohen-addad et al., 2019).
3. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: This algorithm groups data objects based on their density, forming clusters in high-density areas and labeling outliers as noise (Hahsler et al., 2019).
4. **Mean Shift**: This algorithm iteratively moves cluster centers to regions with the highest data density until convergence, thus identifying clusters with irregular shapes (Awangga et al., 2018).
5. **Gaussian Mixture Models (GMM)**: GMM assumes that data objects are generated from a mixture of Gaussian distributions. This algorithm estimates the parameters of these distributions to identify underlying clusters (Patel & Kushwaha, 2020).

The list of algorithms above is just a few examples; in its development up to the present, there have been numerous other clustering algorithms, each with its own assumptions and characteristics. The choice of algorithm depends on the specific problem and the nature of the data being used.

Clustering Benefits

Clustering offers various benefits and can be applied in diverse fields. Some areas that can gain advantages from clustering include:

1. **Business and Marketing Analysis**: Clustering is widely used in business and marketing to segment customers based on their preferences, behavior, or demographics. This allows for targeted marketing strategies, personalized recommendations, and better understanding of customer needs and preferences (Chen et al., 2021; Täuscher & Laudien, 2018).
2. **Image Analysis and Document Processing**: Clustering is applied in image processing and computer vision, such as image segmentation and object recognition. Additionally, clustering can be used for document clustering, topic modeling, and text categorization in natural language processing applications (Cozzolino & Ferraro, 2022; Ji et al., 2019).
3. **Financial Analysis**: Clustering can be employed in financial analysis to identify patterns and groups of stocks, portfolios, or financial market behavior. This aids in investment decision-making, risk management, and market understanding (Iorio et al., 2018).
4. **Genetic and Biology**: Clustering is used to group genetic sequences, gene expression profiles, or other biological data. This assists in understanding genetic and phenotypic relationships, identifying population groups, or classifying types of diseases (Tonkin-Hill et al., 2018).
5. **Security and Anomaly Detection**: Clustering can be used to detect anomaly patterns in security data, such as unusual network attacks or suspicious behavior in transaction data. This supports security monitoring and intrusion detection systems (Pu et al., 2021).

6. **Geo-Spatial Clustering**: In geospatial analysis, clustering can group similar geographic locations based on data like GPS coordinates or other spatial attributes. This aids in understanding spatial patterns, clustering geographic areas, or determining optimal routes (Rahman, 2022).
7. **Pattern Recognition and Data Compression**: Clustering algorithms can be applied in pattern recognition, where we attempt to identify and classify similar patterns or shapes. Additionally, clustering techniques like vector quantization have been used for image data compression, reducing storage and transmission requirements for image data (Du et al., 2022; Sun et al., 2019).

The list above is just a few examples of fields that can benefit from clustering. In reality, the benefits of clustering are not limited to that list. Clustering has broad applications and can offer valuable insights across various domains where data is collected and analyzed.

Clustering for Non-Technical Users

In general, performing clustering requires a basic understanding of statistics and programming. Clustering can be challenging for non-technical individuals who lack knowledge of both statistics and programming due to several reasons:

1. **Selection of Appropriate Methods and Parameters**: Clustering involves selecting a suitable clustering method based on the data and analysis objectives. There are various clustering methods with different assumptions and characteristics, and choosing the wrong one can result in irrelevant or uninformative clustering outcomes. Additionally, parameters such as the number of clusters or distance metrics need to be adjusted, and these should be understood and chosen wisely. Without an understanding of these methods and parameters, it's difficult to achieve optimal clustering results.
2. **Interpretation of Results and Validity**: After performing clustering, it's important to correctly interpret the results. This involves understanding what each cluster represents, how to compare clustering quality between different methods, and how to evaluate clustering validity. Without adequate statistical understanding, it's challenging to interpret and use clustering results properly.
3. **Data Preprocessing and Adjustment**: Often, data preprocessing is required before clustering, involving steps like handling missing values, scaling adjustments, feature selection, or dealing with inappropriate data. Understanding these data preprocessing techniques is crucial to prepare data correctly before clustering. Otherwise, clustering outcomes might be influenced by unstructured data or irrelevant information.
4. **Algorithm Provision and Implementation**: Implementing clustering algorithms involves programming or using specialized software that requires sufficient technical knowledge. Even if user-friendly data visualization tools exist, understanding programming and having the ability to use such software might be necessary for data preparation, parameter configuration, or modifying analyses as needed.
5. **Evaluation and Troubleshooting**: When performing clustering, challenges or issues often arise, such as unstructured data, high noise, or unclear cluster count. Statistical and programming knowledge enables someone to comprehend emerging problems, employ appropriate evaluation techniques, and seek suitable solutions.

Overall, an understanding of statistics and programming plays a crucial role in selecting appropriate clustering methods, interpreting clustering outcomes, handling data preprocessing, and evaluating clustering quality. Without sufficient understanding, clustering can be challenging to perform effectively and obtain meaningful results (Cohen-Addad et al., 2021; Min et al., 2018).

3.0 METHODOLOGY

In this research, the system requirements we compiled were created in accordance with the IEEE 830 SRS standard. The IEEE Software Requirements Specification (SRS) standard, or IEEE 830, is an internationally recognized guideline for documenting software requirements (Amalia et al., 2022). This document aims to communicate clearly and comprehensively what is expected from the software to be developed, both by developers and stakeholders who will use the product. The IEEE SRS standard encompasses various essential elements, such as a general system description, functional and non-functional requirements, as well as constraints governing the software. This document also elucidates the system's interactions with users and the environment, while detailing the software's configurations and dependencies on external factors.

This standard serves not only as a guide for the development team but also as a formal contract between the software requester and the developing party. This ensures that expectations, limitations, and responsibilities related to the software development project are comprehensively and accurately delineated. The IEEE SRS document also aids project management by providing a consistent and structured guideline for the entire team. The use of the IEEE SRS standard in software development offers significant benefits by reducing ambiguity, avoiding misinterpretation, and enhancing transparency and accountability within the project (Ali et al., 2018). By adhering to these guidelines, the development team and stakeholders can achieve a shared understanding of the project's objectives and scope, ultimately contributing to the successful development of efficient and high-quality software.

SRS Introduction

1. **Purpose**: The system within the scope of this SRS is a web-based application designed to assist non-technical users in performing clustering more easily without requiring in-depth knowledge of programming and/or statistics. This system is not confined to specific contexts, meaning that any tabular data (in the form of tables: rows and columns) can be used as long as it adheres to the data format.
2. **Scope of the Problem**: The system within the scope of this SRS must be capable of aiding non-technical users from fields including but not limited to business, marketing, retail, and finance, in conducting and benefiting from data clustering. The system should provide users with the facility to upload data they wish to cluster, select clustering columns/attributes, perform data numbering/vectorization, and create clusters using algorithms that can be easily chosen by the user.
3. **General Description of the Document**: This SRS is a part of the research article manuscript, where its sections are consolidated into a unified whole within the article as a primary component of the research methodology. This SRS is created to provide a clear overview of the characteristics of a system that can assist non-experts in conducting clustering more easily for future research endeavors.

The Global Software Description

1. **Product Perspective**: The product resulting from the implementation of this SRS is a software system that assists non-technical users in performing clustering on any tabular-format data more simply and easily. The implemented system can operate as a standalone web-based application accessible to everyone via the internet, as well as an application that is part of a larger corporate system.
2. **Product Function**: The system within the scope of this SRS encompasses several functions that are generally aimed at aiding the creation of clusters from generic (common) tabular data. When a user employs the system to generate clusters, the system must provide the user with the capability to:
   a. Upload data in the form of a table using CSV or Excel (XLSX) formats. While CSV is recommended as a mandatory supported format due to its easier processing, the Excel format is suggested owing to its widespread use in data analysis.
   b. Select Data, which involves choosing the desired cluster parameters. In tabular data, there can be 1-2 or more columns, not all of which are pertinent or necessary for analysis. Users should be able to select which columns to use as clustering parameters.
   c. Vectorization, which is the process of transforming non-numeric data into numeric data if required.
   d. Create clusters by selecting and/or inputting algorithm clustering parameters.

In an SRS document, the main functions of the application that directly support the achievement of user goals are formulated as functional requirements. The following table summarizes the functional requirements of this clustering system.

Table 1. Functional Requirements

Actor	Description	Code
User	Uploading the dataset to be clustered.	SRS-F-U01
	Selecting specific columns from the dataset to be clustered.	SRS-F-U02
	Converting non-numeric data into numeric if necessary.	SRS-F-U03
	Choosing the clustering algorithm to be used.	SRS-F-U04
	Configuring the parameters of the selected clustering algorithm.	SRS-F-U05
	Executing the clustering algorithm.	SRS-F-U06
	Viewing the resulting clustering diagram and downloading it if needed.	SRS-F-U07

In addition to the core functional requirements as presented earlier, to provide maximum benefits to its users, this clustering system must also satisfy the following non-functional requirements.

Table 2. Non-Functional Requirements

Description	Non-Functional Code
The system must be operational 24 hours a day, 7 days a week.	SRS-NF-N-01

Description	Non-Functional Code
The system must be compatible with commonly used browsers by the general public, namely Google Chrome, Mozilla Firefox, and Microsoft Edge.	SRS-NF-N-02
The system must be accessible using the internet with a bandwidth greater than or less than 3 MB/s.	SRS-NF-N-03
The system's response time must be below 5 seconds.	SRS-NF-N-04
The cluster creation time for data below 10,000 rows should be below 3 minutes.	SRS-NF-N-05
The system should remain user-friendly for users accessing it through browsers on mobile devices.	SRS-NF-N-06

External Interface Requirements

External Interface Requirements are a component of the Software Requirements Specification (SRS) document that elucidates how the software to be developed will interact with other external entities or systems. This encompasses information about how the software will communicate with users, other systems, or external hardware.

1. User Interface: The developed application can implement one or a combination of the following interface styles:
   a. Single-Page Applications (SPAs) offer a unified experience for non-technical users by consolidating all content onto a single page, enabling dynamic interactions without requiring page refreshes. This ensures straightforward navigation, reduces user confusion, and provides a responsive interface.
   b. Card-based design arranges information into distinct cards, facilitating quick comprehension for non-technical users, particularly effective for mobile applications.
   c. Tab-based interface divides related content into sections, enabling users to switch seamlessly between them, encouraging an organized approach to accessing information.
   d. Wizard-style interface guides non-technical users through a series of steps, presenting one step at a time to facilitate understanding and completion of complex tasks.
   e. Accordion menu aids non-technical users in accessing detailed information in a collapsible format, simplifying navigation, and enabling focused exploration.
2. External System Interfaces: The developed clustering system should at least be accessible as a standalone web-based system, reachable by users via the internet. Additionally, the system should be capable of integration with other systems within a company. Integration could involve setting up subdomains under the company's main website domain. Furthermore, deeper integration is preferable, allowing users to choose data for clustering from their repositories and/or corporate databases.
3. External Hardware Interfaces: If the clustering system is implemented as a standalone system, there are no specific requirements in this section. However, if the system is integrated with an institution's/corporation's main system, it should ideally connect to the company's physical server storing the necessary data for analysis. This communication is best conducted directly using LAN protocols to ensure swift data retrieval by the clustering system. Additionally, the clustering system should be operable on hardware with minimum specifications that are not excessively high. At least with constraints as indicated in the following table.

Table 3. Hardware requirements

Hardware	Minimum Specification
Processor	1 GHz x86
RAM	1 GB
Storage	15 GB HDD

4. Communication Protocols: The developed Clustering System must support communication via the HTTP and HTTPS protocols to enable users to access it over the internet. Additionally, if the system is implemented as a subsystem, the communication protocol with the main system's data repository should ideally be LAN.
5. Security Requirements: The clustering system must ensure that user-uploaded data remains inaccessible to third parties. This is crucial, given that user-uploaded data might contain confidential or sensitive information related to privacy concerns. If the system is implemented as a standalone application, user data should be promptly deleted from the server once cluster visualizations have been downloaded. Conversely, if the application is integrated into a corporate system, it must be developed to adhere to or exceed the security standards of the parent system.

Software Functional Description

The primary purpose of the system within the scope of this SRS is to assist users in creating cluster visualizations from input data. Consequently, the key features present in the system should come together as a cohesive unit that effectively aids users in achieving their goals. The following Activity Diagram illustrates the overall user interaction flow with the system within a single cycle.

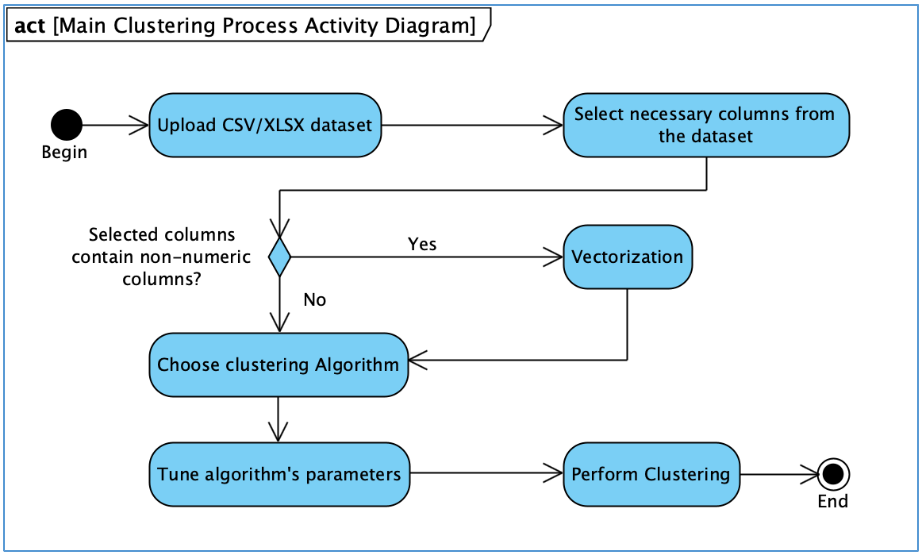

Figure 1. Activity diagram of clustering process

Initially, users who wish to perform clustering upload a dataset to the system. This dataset can be in the form of a CSV or Excel (XLSX) file, formatted in such a way that it is tabular (with rows and columns). Once uploaded, the data is displayed within the system. From there, the user selects the columns they wish to analyze. Afterward, if any of the selected columns contain non-numeric data, the user can perform vectorization, where non-numeric data is transformed into numeric data. During this process, the system displays the selected columns along with their distinct values. The user then assigns appropriate numeric values to each of these distinct values. Once the value assignment is complete, the user can choose a desired clustering algorithm and proceed to adjust the algorithm's parameters. These parameters are displayed according to the selected algorithm. After parameter adjustment is finalized, the user can click a button, and the system will cluster the data. Upon completion of the clustering process, a visualization diagram of the clusters will be presented.

System Architecture

To enhance the effectiveness of this clustering application, we propose a high-level system architecture that prioritizes ease of implementation. In this general system architecture, the system is divided into several components including the Web Server, Request Handler, Model, and Clustering Engine. Users upload datasets in the form of Excel or CSV files through the front-end accessed via common browsers such as Chrome, Firefox, and Edge. These requests are then managed by the web server, for which we suggest using a Python-based web server/framework like Django or Flask. The choice of Python as the programming language for this clustering application takes into consideration the availability of comprehensive and widely-used clustering libraries such as Sci-kit Learn, NumPy, Pandas, and Matplotlib. These libraries will facilitate the implementation of the system, eliminating the need to create clustering algorithms, data conversion, and visualization from scratch.

Upon receiving a request with a dataset file payload, the web server forwards the request to the Request Handler or Controller module of the Clustering application. The Controller module is responsible for recognizing incoming requests and determining the appropriate actions to be taken next. For the upload process, the request is then passed to the Model component. In this component, the file data is converted into SQL queries, which are executed to store the contents of the file in a temporary database table. This storage is temporary and serves to expedite the subsequent clustering process. Specifically, we recommend using a relational database for this purpose, as it simplifies the conversion and temporary storage of data from user-provided Excel/CSV files. Naturally, converting table-format data from files into SQL Inserts in a relational database maintains a seamless transition.

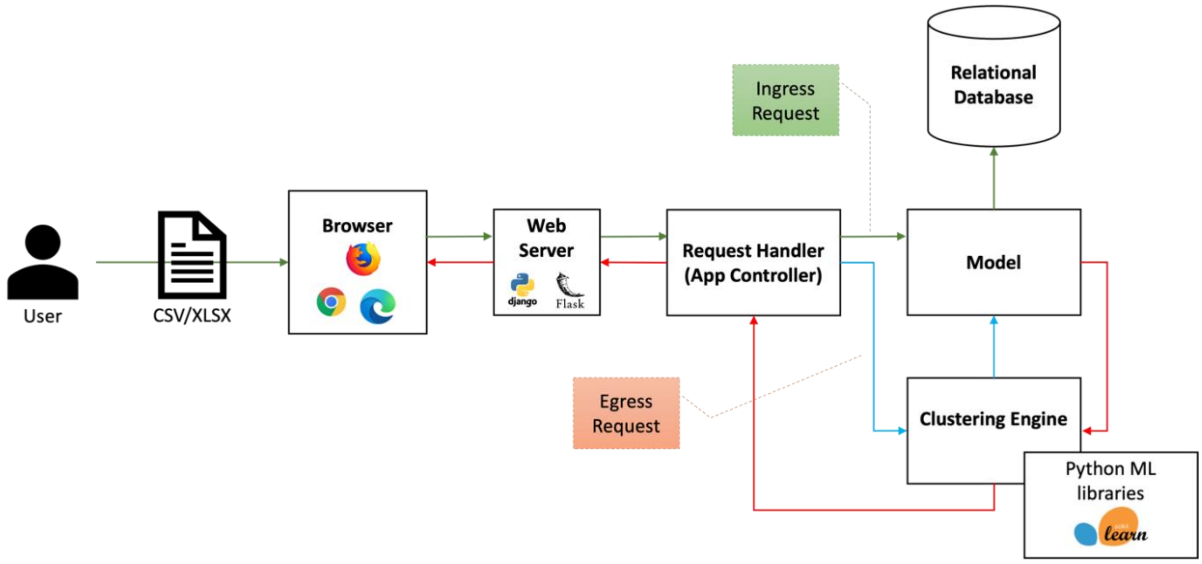

Figure 2. High-level system architecture

Subsequently, when a user initiates the cluster creation process, the request directed to the Web Server is then routed to the Clustering Engine component. This component is responsible for interpreting the selected clustering algorithm input along with its parameters. The Clustering Engine component subsequently retrieves the user-uploaded file data stored in the database through the Model component. Once the data is obtained, the Clustering Engine component invokes the appropriate clustering function within the Python Clustering library, specifically Sci-kit Learn, utilizing the parameters inputted by the user. The resulting clustering outcomes are then transformed into visualizations using charting libraries like Matplotlib. These visualizations are then sent back to the Controller, passed to the Web Server, and finally presented in the user's browser.

4.0 RESULTS AND DISCUSSION

From the summarized SRS, we derived the system's design in the form of a low-fidelity mockup. This design aims to provide a clearer understanding for anyone who will implement this system. The user interface design in this SRS is created utilizing the UI Wizard and Accordion styles. The first page is the initial page where users should find the facility to upload their dataset. This facility consists of a file picker, which is readily available in HTML5. It would be even better if users could directly drag and drop files from their file explorers to the system. At the bottom of the page, there is a "Next" button that remains inactive until data is uploaded.

When the data has been uploaded, a portion of the stored data in the system will be displayed to the user. This is done to confirm that the data uploaded by the user has been successfully stored temporarily in the system. At the same time, the "Next" button that was previously inactive changes its color and is now clickable by the user.

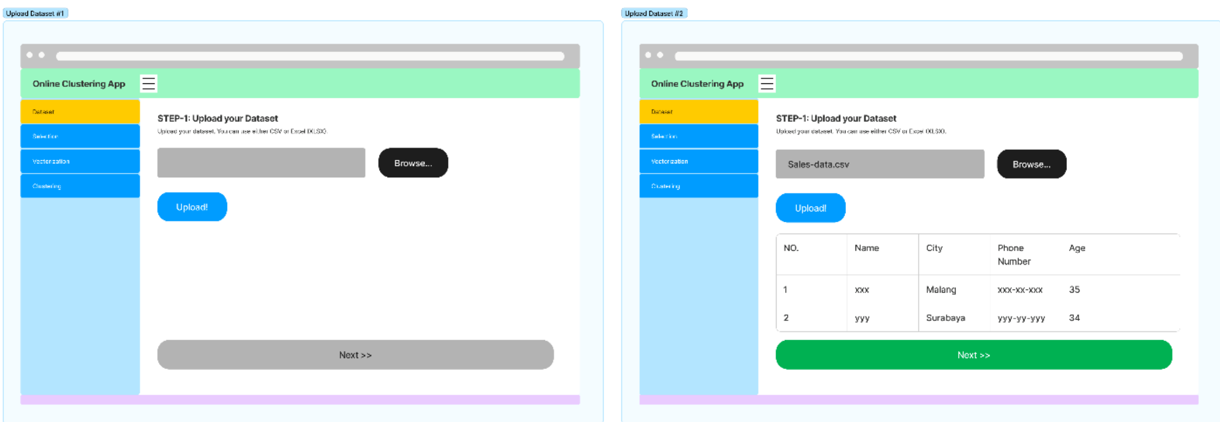

Figure 3. Upload Dataset page

When the user clicks the "Next" button, the system displays step 2, which is the "Data Selection" page. On this page, the columns present in the uploaded data are shown. For each column, the user is provided with the option to select or deselect it. In this design, the built-in HTML5 checkbox component is used. Columns that the user checks with the checkbox will be chosen as the data to be clustered.

Subsequently, when the user clicks the "Next" button again, the page transitions to step 3, the "Vectorization" page. On this page, the columns previously selected by the user along with their distinct values are displayed. The chosen columns are presented using the Accordion interface style. Each accordion contains distinct values from a single column, within which textbox components are provided to input numeric values for those distinct values.

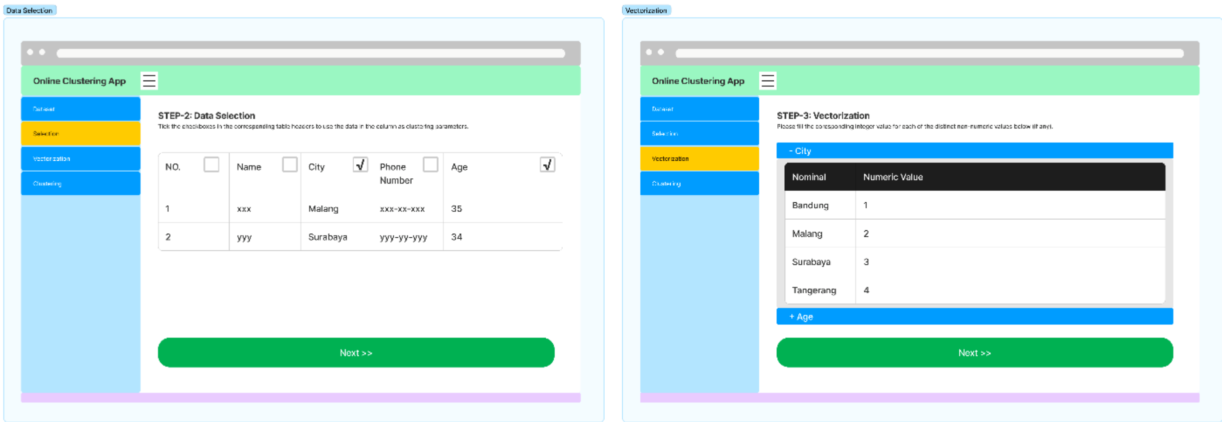

Figure 4. Data Selection and Vectorization pages

After vectorization is completed, if the user clicks the "Next" button again, the page transitions to step 4, which is the clustering page. On this page, the user is provided with a Combo Box to select the desired clustering algorithm. When the user chooses a clustering algorithm, the system displays a form to input the parameters specific to the chosen algorithm. For instance, in this design, if the user selects the K-Means algorithm, the parameter inputs displayed would include at least "number of clusters" and "initial seed", as these are specific parameters for that algorithm.

In the end, after the user has set the parameters for the clustering algorithm, the button at the bottom of the page changes its caption to "Proceed", and upon clicking it, the clustering process is initiated. Once the clustering process is completed, a visual representation of the cluster results in the form of a scatter plot is displayed at the bottom of the page, and this visualization can be downloaded by the user. At this point, the clustering process is concluded. If the user wishes to compare with other algorithms, they can directly change their algorithm choice on the same page and click the "Proceed" button again without having to start over from the beginning steps.

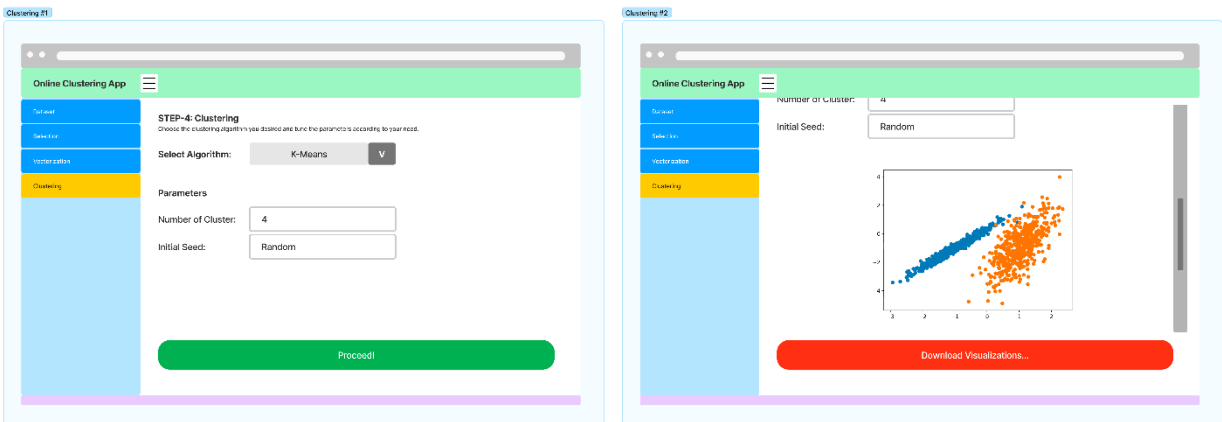

Figure 5. Clustering page

5.0 CONCLUSION

In conclusion, the practice of clustering stands as an integral facet of data analysis, despite often posing challenges for individuals lacking technical expertise. This research encapsulates a comprehensive examination of the literature, delving into the requisites for a clustering application tailored to non-technical users, adhering to the established IEEE 830 SRS standard. Through this meticulous analysis, a fully-fledged SRS has been meticulously formulated, complemented by a robust system architecture and a thoughtfully designed user interface.

The culmination of these efforts is projected to furnish forthcoming researchers and invested stakeholders with an unambiguous and all-encompassing blueprint, aiding their pursuit of developing and effectively implementing this pioneering system. Thus, this research contributes substantively towards the endeavor of simplifying the utilization of clustering methodologies for the wider public, effectively bridging the gap between intricate technical intricacies and the demand for user-friendly accessibility.

Acknowledgement

This research is funded by the Center of Research and Community Service, State Polytechnic of Malang with the Research Grants No. SP DIPA 023.18.2.677606/2022.

References

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics, 9*(8), 1295. https://doi.org/10.3390/electronics9081295

Ali, S. W., Ahmed, Q. A., & Shafi, I. (2018). Process to enhance the quality of software requirement specification document. *2018 International Conference on Engineering and Emerging Technologies (ICEET)*, 1–7. https://doi.org/10.1109/ICEET1.2018.8338619

Amalia, E. L., Lestari, V. A., Wijayaningrum, V. N., Abdullah, M. Z., Ardyaningrum, N. A., & Safitri, W. (2022). Software Requirement Specification for Learning Man-agement System Design. In R. A. Asmara, A. R. Syulistyo, V. N. Wijayaningrum, M. S. Khairy, I. Siradjuddin, & S. E. Sukmana (Eds.), *Proceedings of the 2022 Annual Technology, Applied Science and Engineering Conference (ATASEC 2022)* (pp. 145–158). Atlantis Press International BV. https://doi.org/10.2991/978-94-6463-106-7_14

Awangga, R. M., Pane, S. F., Tunnisa, K., & Suwardi, I. S. (2018). K Means Clustering and Meanshift Analysis for Grouping the Data of Coal Term in Puslitbang tekMIRA. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 16*(3), 1351. https://doi.org/10.12928/telkomnika.v16i3.8910

Bjälkebring, P. (2019). Math Anxiety at the University: What Forms of Teaching and Learning Statistics in Higher Education Can Help Students With Math Anxiety? *Frontiers in Education, 4*, 30. https://doi.org/10.3389/feduc.2019.00030

Chen, M.-K., Wu, C.-M., Chen, L.-S., & Huang, Y.-P. (2021). The Influential Factors of Taiwan SMEs' Clustering Keystone Business Strategy—The Perspective of Business Ecosystem Using FAHP. *Sustainability, 13*(18), 10304. https://doi.org/10.3390/su131810304

Cohen-addad, V., Kanade, V., Mallmann-trenn, F., & Mathieu, C. (2019). Hierarchical Clustering: Objective Functions and Algorithms. *Journal of the ACM, 66*(4), 1–42. https://doi.org/10.1145/3321386

Cohen-Addad, V., Saulpic, D., & Schwiegelshohn, C. (2021). A new coreset framework for clustering. *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 169–182. https://doi.org/10.1145/3406325.3451022

Cozzolino, I., & Ferraro, M. B. (2022). Document clustering. *WIREs Computational Statistics, 14*(6), e1588. https://doi.org/10.1002/wics.1588

Du, F., Li, X., Li, D., & Shen, W. (2022). Self-adaptive real-time clustering analysis and damage pattern recognition for steel tube confined reinforced concrete structures through acoustic emission signals. *Structural Control and Health Monitoring, 29*(11). https://doi.org/10.1002/stc.3071

Fergana. (2021). Light Industry Enterprises In Marketing Activities Experience Of Foreign Countries In The Use Of Cluster Theory. *The American Journal of Management and Economics Innovations, 03*(01), 36–39. https://doi.org/10.37547/tajmei/Volume03Issue01-08

Garcia-Dias et al. - 2020—Clustering analysis.pdf. (n.d.).

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). dbscan: Fast Density-Based Clustering with R. *Journal of Statistical Software, 91*(1). https://doi.org/10.18637/jss.v091.i01

Iorio, C., Frasso, G., D'Ambrosio, A., & Siciliano, R. (2018). A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications, 95*, 88–103. https://doi.org/10.1016/j.eswa.2017.11.031

Ji, X., Vedaldi, A., & Henriques, J. (2019). Invariant Information Clustering for Unsupervised Image Classification and Segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9864–9873. https://doi.org/10.1109/ICCV.2019.00996

Kansal, T., Bahuguna, S., Singh, V., & Choudhury, T. (2018). Customer Segmentation using K-means Clustering.

Kranzler, J. H., & Anthony, C. J. (2022). *Statistics for the Terrified*. Rowman & Littlefield.

Lecturer Fergana State University, Republic Of Uzbekistan, Fergana, Zokirova, S. X., Ahmedova, D., Lecturer Fergana State University, Republic Of Uzbekistan, Fergana, Akbarov, R. F., Lecturer Fergana State University, Republic Of Uzbekistan, Fergana, Xonkeldiyeva, K. R., & Lecturer Fergana State University, Republic Of Uzbekistan,

Linares-Mustarós, S., Coenders, G., & Vives-Mestres, M. (2018). Financial performance and distress profiles. From classification according to financial ratios to compositional classification. *Advances in Accounting, 40*, 1–10. https://doi.org/10.1016/j.adiac.2017.10.003