Implementation of K-Means and K-Nearest Neighbor Methods for Laptop Recommendation Websites

Vincentius Riandaru Prasetyo[1*], Mohammad Farid Naufal[2], and Budiarjo[3]

[1,2,3] Department of Informatics Engineering, Faculty of Engineering, University of Surabaya, Indonesia
*vincent@staff.ubaya.ac.id

Abstract. Along with technology development, laptops are becoming increasingly popular and are handy tools in everyday life. However, with so many brands and laptops available, people often find it difficult and need help choosing the laptop that best suits their needs and desires. A website-based system has been created to provide laptop recommendations based on user needs and preferences. This system uses the K-Nearest Neighbor (KNN) method to classify user input with datasets that have been grouped using the K-Means method. Thus, users can choose the right laptop according to their needs with the help of this system. Based on the tests' results, the highest accuracy in the training process is 97%, with a total dataset of 1000 data, which comes from the websites versus.com and arenalaptop.com. Whereas for the validation results obtained from 51 users, the majority stated that the recommendation results were by the criteria of the users.

Keywords: laptop, recommendation, K-Means, KNN.

1 Introduction

Global statistical data shows that 161.6 million laptops were sold, 65.23% more than desktop PCs and nearly the same number as tablets in 2017 [1]. Using computers in the work environment has dramatically helped and made it easier for people to complete tasks. However, recently, laptops have become preferable to desktop computers. The increasing use of laptops is in line with the increasing mobility of people in carrying out their activities. In addition, laptops continue to develop their technology with various specifications to become an attraction for consumers [2]. Many consumers prefer laptops for their capabilities, portability, and mobility compared to desktop PCs. In the market, many laptops have various brands and features. This makes the selection of a laptop that fits the buyer's needs essential [3]. In the laptop selection process, many consumers find it challenging to choose the right laptop because all laptops look the same [4]. Therefore, this study aims to develop a recommendation system for selecting laptops based on brand attributes, price, processor (CPU), graphics card (VGA), RAM, internal memory, screen size, and screen resolution. The built system implements the K-Means and K-Nearest Neighbor (KNN) methods.

M. Hartono et al. (eds.), *Proceedings of the 4th International Conference on Informatics, Technology and Engineering 2023 (InCITE 2023)*, Atlantis Highlights in Engineering 21,
https://doi.org/10.2991/978-94-6463-288-0_38

K-Means is a data grouping method included in the unsupervised learning method. This method will group data by system partition. K-Means aims to minimize data variations in a cluster. The main principle of this method is to arrange the cluster center (centroid) of the dataset [5]. In this study, the K-Means method was used to group laptop data that is similar or highly similar. This study used the K-Means method to assist in the data labeling process. The results of the clustering process with K-Means will produce different clusters, where keywords will be taken from each cluster member data to determine the cluster's label.

For example, the first cluster has 100 data as its members, then from these 100 data, keywords will be taken for each data to be further categorized into one of the categories. For example, if there are keywords such as "High VGA" and "High RAM", then the data is categorized under "Gaming". After the 100 data get categories, the system will determine which category is the most dominant of the 100 data. For example, if 70 out of 100 tweets are categorized as "Gaming", all data in the cluster will be labeled "Gaming". This process applies to other clusters. The results of data labeling using the K-Means method will be used by KNN as training data to form a classification model. The KNN method classifies an object based on learning data with the closest distance to the object. This approach is used to find output by comparing the weights of several features. In KNN Classification, test data, and training data will be processed to determine their similarities and draw conclusions based on the results of these comparisons [6].

Research on making a laptop recommendation system has been done before. Research conducted by Ayundita et., al [7] developed a recommendation system for laptops using a recommendation system based on the CRS framework from previous consumer conversations. The accuracy obtained from testing based on system performance for the recommendation system is 84.6%. Based on user satisfaction, the recommendation system built has a higher percentage than the recommendation system found in several e-commerce based on six factors, namely trust, ease of use/usability, ease of understanding, perceived quality of recommendations, information, and perceived efficiency. Another study conducted by Goswami and Behera [8] aims to select the best laptop model among the six models available in the market. After looking at customer ratings, six laptop models were selected from various online shopping sites with different specifications. This research applies the Multiple-Criteria Decision-Making (MCDM) method to recommend the best laptop among the six models. The selection process was based on seven criteria: processor, hard disk capacity, operating system, RAM, screen size, brand, and color.

In particular, the development of a laptop recommendation system by implementing the KNN method was carried out by Rahardja et al. [9]. This system is designed to help users choose a laptop that suits their purchase purpose, such as for design, office, gaming, and others. This system provides recommendations to users to decide which laptop fits their needs. The results of measuring user satisfaction show that 8 out of 10 users agree with the recommendations. Therefore, the level of user satisfaction with the recommendation results reaches 80%, indicating the success of the laptop recommendation system using the KNN method. Tanoko has carried out another development on a laptop recommendation system using the same method [10], where the system built is an Android-based chatbot application. The chatbot application can display recommendation results based on brands, criteria, and prices received by the system through previous conversations. Based on tests carried out by validating 17 random users, the average user is satisfied with the recommendations given, with a satisfaction percentage of 70.6%.

Unlike previous studies, this research adds the K-Means method to group datasets to automatically produce clusters according to the needs and uses of laptops. In addition, this study also aims to measure the accuracy of several combinations of scaling types, namely MinMax Scaler, Standard Scaler, and Normalizer, with distance calculations used in the K-Means and KNN methods, namely Euclidean, Manhattan, and Cosine. This is done to determine which combination is best to implement in the system being built.

2 Research Methods

2.1 Dataset Collection

The laptop data and specifications dataset will be collected through crawling from the versus.com and arenalaptop.com websites. Crawling is a technique for taking web page components and extracting the components contained therein. Determining a web URL is the first step in the crawling process. The URL is then downloaded to get the page set from that URL. The page obtained then extracted information related to the needs [11].

The versus.com and arenalaptop.com websites are used because they have pretty complete information and a relatively simple HTML structure so that the crawling process can be carried out efficiently. The information that will be retrieved on this website is the laptop name, brand, picture, price, and laptop specifications, including processor (CPU), graphics card (VGA), RAM, internal memory, screen size, and screen resolution. In addition to this information, criteria for laptop needs such as gaming, design editing, office, or daily needs will also be taken through a crawl process on the arenalaptop.com website. The library that will be used for crawling, BeautifulSoup, is a Python data extraction library designed by Leonard Richardson and other open-source developers. This library can process HTML and XML documents and provides simple methods for interacting with the DOM model. [12].

2.2 Preprocessing

Preprocessing converts raw data into data easily interpreted by machine learning models and will be easier to use in the training process [13]. The preprocessing process will be feature extraction in the form of encoding labels and data scaling. Label encoding aims to convert text data into numeric data that a machine can read. The label encoding process will read each data in the selected column/feature and produce a different or unique numeric value; then, each value will be entered into a data map containing the numeric value of each label [14]. In this system, the encoding label will be applied to laptop features or specifications in text, such as CPU, graphics card, CPU speed, RAM, internal memory, and screen resolution. The encoding label will be applied to RAM and internal memory due to variations in the size of the limited RAM and internal memory, as well as differences in memory units due to the memory using megabytes (MB) and gigabytes (GB).

The second preprocessing is data scaling. Data scaling aims to make variable values in the same range and prevent one variable from dominating [15]. In this system, the types of scaling data used are MinMax Scaler, Standard Scaler, and Normalizer; these three scalers will be used to see the highest accuracy produced. MinMax Scaler is one of the most widely known approaches to standardizing information. MinMax scaler converts a data series into a range of 0 to 1. The advantage of using MinMax Scaler is that after changing the data, MinMaxscaler retains the shape of the original distribution and does not change the information embedded in the original data [16]. The MinMax Scaler calculation can be seen in equation (1), where *x'* is the new data and X_i is the *i*-th data.

$$x' = \frac{X_i - min\,(X)}{max(X) - min(X)} \tag{1}$$

The second scaling method is the Standard Scaler. A standard scaler converts a series of data into new data with an average value of 0 and a standard deviation of 1 [17]. The calculation of the Standard Scaler can be seen in equation (2), where *x'* is the new data, X_i is the *i*-th data, μx is the average of the data *x*, and σx is the standard deviation of *x*.

$$x' = \frac{X_i - \mu X}{\sigma X} \tag{2}$$

The last scaling method tested is Normalizer. The Normalizer changes a series of data into a new data form with a unit norm; by default, the Normalizer uses the L2 norm (Euclidean norm), where a new data series, if calculated using the Euclidean formula, will be equal to 1. Calculation of the L2 Norm Normalizer can be seen in equation (3), where *x'* is the new data and X_i is the *i*-th data.

$$x' = \frac{X_i}{\sqrt{\sum_{i=0}^{i=n}(Xi^2)}} \tag{3}$$

2.3 Model Training

After preprocessing, the dataset will be grouped using the K-Means method after preprocessing, producing several clusters/groups for forming a classification model. The K-Means method is the most well-known and widely used clustering method compared to other clustering methods. The K-Means method belongs to unsupervised learning algorithms, which can classify data without prior training and without knowing the labels or predetermined results. This method groups data based on each data's similarity [18].

The initial step in the clustering process using the K-Means method is to determine the number of clusters (K) that are the target and the number of centroids needed. Centroid is the central location of each cluster. K-Means will group each data closest to the centroid, where several centroids are randomly selected beforehand. The centroid position will change based on the average of each data in the cluster, and distance calculations are carried out to form a new cluster. The iteration will stop if there is no change in the value/position of the centroid or iterations have reached the specified number [19]. An illustration of the results of the grouping process with the K-Means method is shown in Fig. 1.

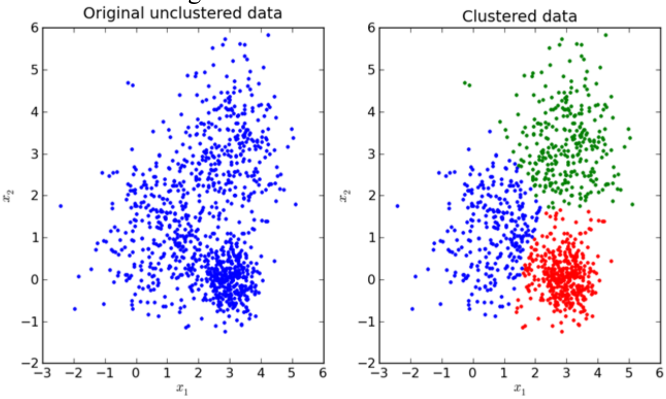

Fig. 1. The Illustration of Clustering

After grouping the datasets with K-Means has been completed, the following process is forming a classification model using the KNN method. The KNN is a classification method that belongs to non-parametric algorithms, which do not care about or make assumptions about data distribution patterns. Therefore, KNN is more relevant in solving real-world problems where data only sometimes follows theoretical assumptions. In addition, KNN has a faster training time and calculates the distance to all data in the dataset [20]. KNN classifies an input based on the number of K nearest neighbors, calculated by distance. For example, in Fig. 2, each data is represented as a dot, with the gray dot representing new data whose class is unknown. In cases where K=4, KNN will look for the four closest points to the gray points and determine the class of gray points based on the majority of the closeness of the closest data. [21].

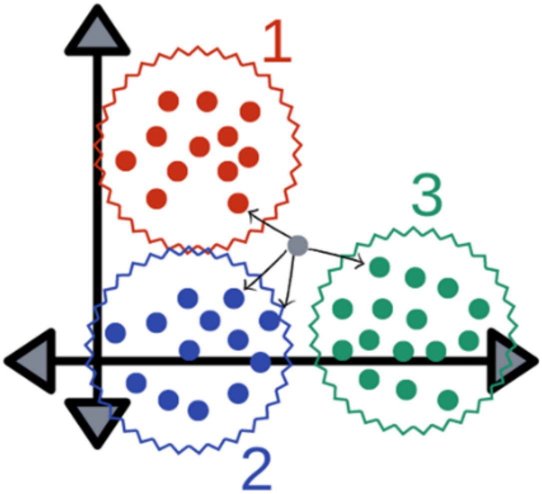

Fig. 2. The Illustration of Classification Using KNN

Determining the K value in the KNN method is very important because a K value that is too small can cause noise which significantly affects the classification results. In addition, a small K value can also cause overfitting, where the model performance is poor on data testing. Conversely, if the value of K is too large, the training process will take longer, and the computational process will become heavier. K values that are too large can also cause underfitting models, where the model performance is poor on training and testing data [22]. In general, determining the value of K in the KNN method is calculated using equation (4), where n is the amount of training data.

$$k = \sqrt{n} \tag{4}$$

Measuring distance or distance is essential to determine the level of similarity or similarity between data. The goal is to determine whether the data is related or similar. Distance measurement is essential in the K-Means and KNN methods because it is needed to help determine groups based on the shortest distance [23]. In this study, the distance measurement methods being compared are Euclidean, Manhattan, and Cosine. Euclidean Distance is the distance between two points in a straight line. This distance method uses the Pythagorean theorem and is the most commonly used distance measurement in machine learning processes [24]. The Euclidean Distance formula is the square root of the difference between two vectors, as shown in equation (5).

$$d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - y_{jk})^2} \tag{5}$$

In this equation, d is the distance between points i and j, where i is the cluster data center with j data on k attributes. The second distance measurement method used is the Manhattan Distance. Manhattan Distance, or "city block distance," is the sum of the distances between all data attributes. For two data points, x, and y, in d-dimensional space, the distance between these points is defined by equation (6). [25].

$$d_{ij} = \sum_{k=1}^{n} |x_{ik} - y_{jk}| \tag{6}$$

Cosine distance is a method that is often used to deal with high dimensional problems at Euclidean distance. The cosine distance is based on the cosine of the angle between two points. If two points have the same orientation, the value is 1, whereas if they are opposite, the value is -1. [26]. Equation (7) shows how to calculate the distance between 2 points using the cosine distance method, where d is the distance between i and j, i is the cluster data center, j is data on attributes, k is symbols of each data.

$$d_{ij} = \frac{\sum_{k=1}^{n}(x_{ik} \cdot y_{jk})}{\sqrt{\sum_{k=1}^{n}(x_{ik}^2) \cdot \sum_{k=1}^{n}(y_{jk}^2)}} \tag{7}$$

2.4 Model Training

At the implementation stage, the programming languages used are PHP and Python. PHP is used to implement the user interface display or user interface. In contrast, Python is used for crawling, model training, and calculating the accuracy of the classification model. The two main Python libraries used in system implementation are BeautifulSoup which is helpful for the crawling process, and scikit-learn, which is used for the model training process and calculating model accuracy.

Scikit-learn is the most comprehensive and open-source library for machine learning in Python. Scikit-learn has many features that make it stand out among other machine-learning software. The implementation of machine learning algorithms in Scikit-learn is optimized for computational efficiency. Another advantage of Scikit-learn is its strong community support for documentation, bug tracking, and quality assurance. Scikit-learn has a fixed model fitting procedure, making switching from one method to another easy [27].

2.5 System Evaluation

At this stage, evaluation is carried out in two ways: measuring the accuracy of the classification model produced in the training process and user validation. Accuracy measurement begins by dividing the existing dataset into two parts, namely 70% as training data and 30% as testing data. After that, it will be calculated how much testing data is classified correctly based on the classification model generated from the training process using 70% of the existing training data [28].

User validation is done by asking several respondents to try the recommendation system that has been built. After that, respondents were asked to provide an assessment of the experiences they had while using the recommendation system. In addition to providing an assessment, respondents can also provide criticism and suggestions for the development of a recommendation system in the future.

3 Results and Discussions

The recommendation process begins with the user selecting the desired laptop needs: gaming, design editing, office, or daily needs. Users can also fill in the desired laptop specifications. After that, the system will collect datasets related to various types of laptops according to previous user input. The dataset used in the built system was obtained through a crawling process on versus.com and arenalaptop.com. After crawling, the dataset will be stored in the CSV file for further processing. The dataset obtained is 1000 data, with each data having ten features, where these features will be used in calculating the K-Means and KNN methods. The ten features are laptop name, brand, picture, price, processor (CPU), graphics card (VGA), RAM, internal memory, screen size, and screen resolution. The number of classes used for the classification process is 4, namely gaming, design editing, office, or daily needs. The amount of data for each class depends on the clustering results performed by the K-Means method. The labeling results carried out by the K-Means method have been validated by Anang Wahyudi as Director of CV Fasterindo Dotcom and laboratory assistant at the Department of Informatics Engineering, University of Surabaya.

The following process to be carried out is preprocessing, as explained in the previous sub-chapter. After preprocessing is done, the following process is model training. The training process begins with the clustering stage using the K-Means method. The K-Means method will first find the most optimal K value using the elbow method. Based on Fig. 3, the most optimal number of K is 4. After getting a dataset that has clusters, the following process is to classify it with KNN. The classification process will be carried out by trying three types of scaling, namely MinMaxScaler, StandardScaler, and Normalizer, and three distance calculations, namely Euclidean, Manhattan, and Cosine. The results of the accuracy comparison of the combination of each type of scaling and distance can be seen in Fig. 4.

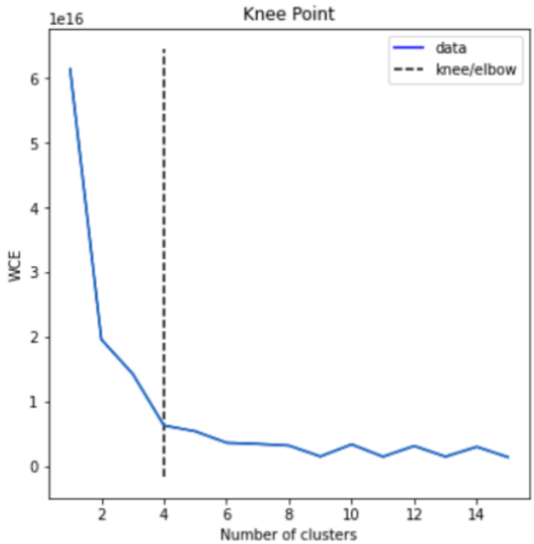

Fig. 3. The Result of Elbow Method in Determining the Number of K in K-Means

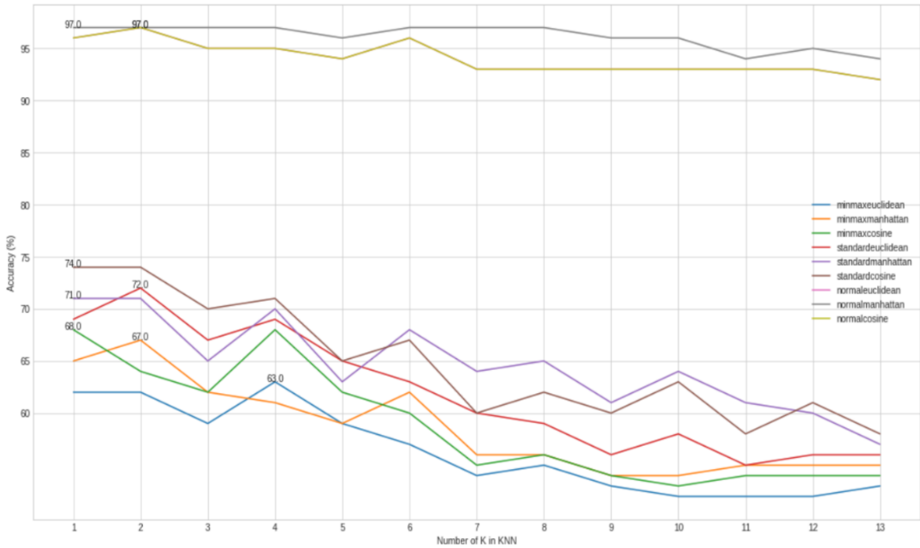

Fig. 4. Comparison of the Accuracy of Applying the K-Means and KNN Method with Several Combinations of Scaling and Distance Methods

Based on the results of the accuracy comparison, it was found that the best combination of scaling and distance is Normalizer and Manhattan distance. This is because this combination has the highest accuracy compared to the other combinations. The accuracy obtained from this combination is 97%, the K value used in the K-Means method is 4, and the K value in the KNN method is 2. Combining the MinMax scaler with several distances produces the highest accuracy of 68% for the Cosine method, and combining the Standard scaler with several distances produces the highest accuracy of 74% using the Cosine method.

In addition to measuring the accuracy of the combination of K-Means and KNN methods, accuracy measurements are also carried out if only using the K-Means method. This measure is done to prove that the combination of K-Means and KNN methods does produce better accuracy. Based on Fig. 5, the highest accuracy of applying the K-Means method alone is 50% with a value of K = 2 for the Normalizer scaling method, and the distance method used is Cosine or Euclidean. The combination of other scaling and distance methods in applying the K-Means method has an accuracy of around 18% - 24%, most of which are at a value of K = 4. Based on the accuracy comparison experiment between the use of a combination of K-Means and KNN methods with the use of the K-Means method alone, the better accuracy is the combination of K-Means and KNN methods.

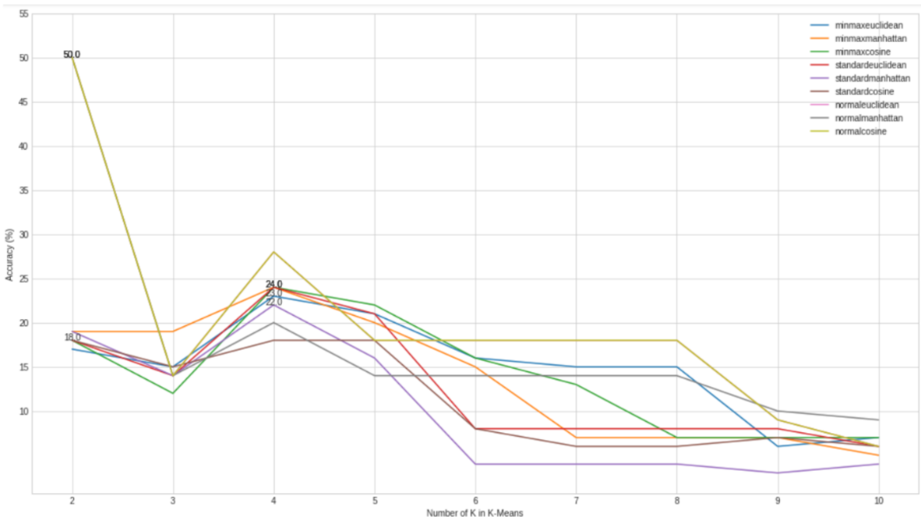

Fig. 5. Comparison of the Accuracy of Applying the K-Means Method with Several Combinations of Scaling and Distance Methods

Another evaluation process is carried out by validating users or respondents. Validation was carried out on 51 respondents to assess whether the results of the system's recommendations were appropriate. At first, users are asked to try the recommendation system that has been made by running all the available features. After that, the user is asked to provide an assessment from 1 to 5 regarding the recommendations provided by the system. The respondents were students, university students, workers who already had laptops, and those who still had problems choosing a laptop. Fig. 6 shows the results of the assessment conducted by 51 respondents, in which 41 respondents (80.4%) answered strongly agree, eight respondents (15.7%) answered agree, and two people (3.9%) answered neutral.

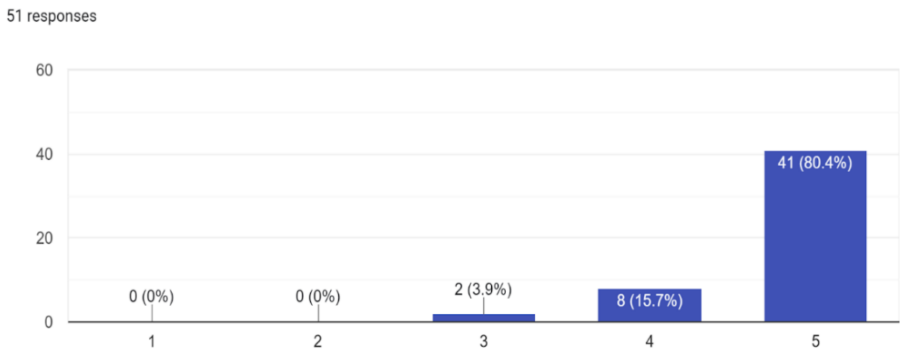

Fig. 6. Results of Assessment by Respondents

In addition to providing an assessment of the recommendations, respondents were also asked to provide input, criticism, and suggestions for the system that had been created. Input provided by respondents included: adding a search button on the "Laptop List" page to make it easier to find laptops, developing the system in the future in a mobile version, and making the system filter laptops with various types of specifications entered by the user.

4 Conclusion

Based on the results of measuring model accuracy and validation of 51 users, the system created has been able to meet user needs in finding a laptop that suits the user's wishes. The results of the laptop recommendations provided by the system are by the needs and specifications entered by the user. In addition, based on testing several scaling and distance combinations, the best combination is obtained from the combination of Normalizer and Manhattan distance with the highest accuracy of 97%.

References

1. Cakir, F. S., Pekkaya, M.: Determination of Interaction between Criteria and the Criteria Priorities in Laptop Selection Problem. International Journal of Fuzzy Systems (5), 1177–1190 (2020).
2. Kusnadi, A., Kurniawan, E.: Implementation of Topsis Method in Web Based System Recommendations for Students Laptop Selection (Case Study: Bhinneka.com). International Journal of New Media Technology (4), 42–45 (2017).
3. Adali, E. A., Isik, A. T.: The multi-objective decision making methods based on MULTIMOORA and MOOSRA for the laptop selection problem. Journal of Industrial Engineering International (13), 229–237 (2017).
4. Lakshmi, T. M., Venkatesan, V. P., Martin, A.: Identification of a Better Laptop with Conflicting Criteria Using TOPSIS. Journal of Information Engineering and Electronic Business (6), 28–36 (2015).
5. Prasetyo, V. R.: Searching Cheapest Product on Three Different E-Commerce Using KMeans Algorithm. In: 2018 International Seminar on Intelligent Technology and Its Application, pp. 239–244. IEEE, Bali (2018).
6. Fitria, H., Wibowo, Indriyani, F.: K-Nearest Neighbor Method for Monitoring of Production and Preservation Information (Treatment) Of Rubber Tree Plant. In: Proceesing of International Conference on Information Technology and Business 4, pp. 29–44. Informatics and Business Institute Darmajaya, Bandar Lampung (2018).
7. Ayundhita, M. S., Baizal, Z. K. A., Sibaroni, Y.: Ontology-based conversational recommender system for recommending laptop. In: The 2nd International Conference on Data and Information Science, pp. 1-8. IOPScience, Bandung (2018).
8. Goswami, S. S., Behera, D. K.: Best Laptop Model Selection by Applying Integrated AHP-TOPSIS Methodology. International Journal of Project Management and Productivity Assessment (9), 2021.
9. Rahardja, C. A., Juardi, J., Agung, A.: Implementasi Algoritma K-Nearest Neighbor Pada Website Rekomendasi Laptop. Jurnal Buana Informatika (10), 75-84 (2019).

10. Tanoko, Y.: Perancangan Aplikasi Chatbot Rekomendasi Laptop dengan Metode K-NN. Universitas Dinamika Bangsa, Jambi (2020).
11. Prasetyo, V. R., Samudra, A. H.: Hate Speech Content Detection System on Twitter using KNearest Neighbor Method. In: International Conference on Informatics, Technology, and Engineering 2021. AIP, Surabaya (2021).
12. Uzun, E., Yerlikaya, T., Kirat, O.: Comparison Of Python Libraries Used For Web Data Extraction. In: International Scientific Conference on Engineering, Technologies and Systems TECHSYS 2018. IOPScience, Plovdiv (2018).
13. Alshdaifat, E., Alshdaifat, D., Alsarhan, A., Hussein, F., El-Salhi, S. M. F. S.: The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance. Data (6), 1-23 (2021).
14. Oktavirahani, F. A., Maharesi, R.: Implementasi Algoritma Decision Tree Cart Untuk Merekomendasikan Ukuran Baju. Jurnal Riset Komputer (9), 138-147 (2022).
15. Radhi, M., Amalia., Sinurat, S. H., Sitompul, D. R. H., Indra, E.: Prediksi Water Quality Index (Wqi) Menggunakan Algoritma Regressi Dengan Hyper-Parameter Tuning. Jurnal Sistem Informasi dan Ilmu Komputer Prima (5), 44-50 (2021).
16. Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., Padma, V.: Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. In: Third International Conference on Smart Systems and Inventive Technology, pp. 729-835. IEEE, Tirunelveli (2020).
17. Pham, D., Kaltenegger, L.: Color classification of Earth-like planets with machine learning. Monthly Notices of the Royal Astronomical Society (504), 6106-6116 (2021).
18. Jahwar, A. F., Abdulazeez, A. M.: Meta-Heuristic Algorithms for K-means Clustering: A Review. Palarch's Journal of Archaeology of Egypt/Egyptology (17), 1-20 (2021).
19. Ahmed, M., Seraj, R., Islam, S. M. S.: The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. Electronics (9), 1-12 (2020).
20. Lee, T. R., Wood, W. T., Phrampus, B. J.: A Machine Learning (kNN) Approach to Predicting GlobalSeafloor Total Organic Carbon. Global Biogeochemical Cycles (33), 37-46 (2019).
21. Priya, B. G.: Emoji Based Sentiment Analysis Using KNN. International Journal of Scientific Research and Review (7), 859-865 (2019).
22. Jhamtani, A., Mehta, R., Singh, S.: Size of wallet estimation: Application of K-nearest neighbour and quantile regression. IIMB Management Review (33), 184-190 (2021).
23. Baldini, G., Geneiatakis, D.: A Performance Evaluation on Distance Measures in KNN for Mobile Malware Detection. In: 2019 6th International Conference on Control, Decision and Information Technologies, pp. 193-198. IEEE, Paris (2019).
24. Suwanda, R., Syahputra, Z., Zamzami, E. M.: Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K. In: 4th International Conference on Computing and Applied Informatics 2019. IOPScience, Medan (2019).
25. Faisal, M., Zamzami, E. M., Sutarman.: Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance. In: 4th International Conference on Computing and Applied Informatics 2019. IOPScience, Medan (2019).
26. Usino, W., Prabuwono, A. S., Allehaibi, K. H. S., Bramantoro, A., Amaldi, W.: Document Similarity Detection using K-Means and Cosine Distance. International Journal of Advanced Computer Science and Applications (10), 165-170 (2019).

27. Hao, J., Ho, T. K.: Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. Journal of Educational and Behavioral Statistics (44), 348-361 (2019).
28. Muraina, I, O.: Ideal Dataset Splitting Ratios In Machine Learning Algorithms: General Concerns For Data Scientists And Data Analysts. In: 7th International Mardin Artuklu Scientific Researches Conference, pp. 496-504. Euroasia Journal, Mardin (2021).