JQCM
University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Applying Collaborative Filtering in E-Marketing Systems

***Kasem Maher Ahmed* [a,*], *Omar Muayad Abdullah* [a]**

[a] *Department of Computer Science, College of Computer Sciences and Mathematics, University of Mosul, Mosul, Iraq. Email: qasim.24csp21@student.uomosul.edu.iq, omaraldewachy@uomosul.edu.iq*

ARTICLE INFO

Article history:
Received: 20/10/2025
Rrevised form: 22 /12/2025
Accepted : 30 /12/2025
Available online: 30/03/2026

Keywords: Machine Learning Algorithms, Collaborative Filtering, K-Nearest Neighbors (KNN) Algorithm, Decision Tree Algorithm.

ABSTRACT

Collaborative filtering is very important in e-marketing systems since it enables users to search through the large product lists and become better buyers. This paper will look into the application of machine learning algorithms for instance K-Nearest Neighbors (KNN) and Decision Tree Classifier in formulating a collaborative filtering for supermarket products like, food, dairy, canned goods, juices and detergents. Three data files were employed; a supermarket products file of five Categories each having ten products, a supermarket manager's offers file of the same Categories with each set with four products, and a file of past user's purchases. The methodology employed involves using a user product matrix to capture purchasing behavior, using KNN algorithms to predict user preferences using the distance of Euclidean, also, a decision tree to set up the decision-making rules for products recommendations implemented. The number of the previous user's purchases is shown in a histogram. The paper compares the KNN algorithm with the decision tree algorithm on recommendation accuracy, precision, recall, F1-score and execution time. The KNN algorithm yielded the best accuracy (0.9651), precision (0.8233), recall (0.8430), F1 score (0.8293) and execution time (0.896). The decision tree algorithm yielded lower accuracy (0.8392), precision (0.2094), recall (0.2111), F1 score (0.2072), and execution time (3.686).

MSC..

https://doi.org/10.29304/jqcsm.2026.18.12391

1. Introduction

Collaborative filtering is very important in present e-marketing systems because it supplies user's specific recommendations according to their tastes and preferences. Collaborative filtering enables customers to find the right product from a wide range of products, effectively allowing them to obtain products [1]. Collaborative filtering provides recommendations of products by identifying specific trends of similarity among users, the K-Nearest Neighbors (KNN) algorithm is used to examine those trends. The performance of this algorithm is based on the fact that, products that consumers purchase are of interest to consumers who have similar purchasing patterns, also the KNN algorithm has the greatest efficacy if it is capable of accessing enough information [2].

E-marketing systems have rendered many products to the users, the input on the one hand giving users a greater choice, but on the other hand bringing them information overkill, because many users find it impossible to search for the right products. Collaborative filtering has evolved to solve this problem, it supplies product

*Corresponding author: Kasem Maher Ahmed

Email addresses: qasim.24csp21@student.uomosul.edu.iq

Communicated by 'sub etitor'

recommendations to users, on the basis of their prior interaction. The use of algorithms based on machine learning is common in collaborative filtering. The k-nearest neighbor (KNN) algorithm is one of the foremost display algorithms on the part of users that are based on similarity of user's behavior, it is used in turn to find out users that have a similar purchasing behavior and recommend to such users, products that their neighbors might buy [3]. The KNN algorithm works, by measuring Euclidean distance between users, in order to ascertain similarity. Nearest neighbors are identified based on this distance, and products recommended based on user's similarity.

The other algorithm is the decision tree which produces a tree-like structure with every node depicting a choice taken by the algorithm on the basis of the patterns of earlier purchases made by users, the branches indicate possible outcomes whereas the leaves give recommendations [5]. A decision tree organizes data in a clear way, so that the tree analyzes the user's past behavior to suggest the most suitable products [6]. This research paper presents the creation of a collaborative filtering for supermarket products, such as Food, Dairy, Canned Goods, Juices, and Detergents.

Three data files are used: a data file for supermarket products, a data file for offers, and a dynamically updated data file for previous users. The system provides personalized recommendations based on KNN and decision tree algorithms, displaying the number of users' purchases in a histogram, indicating the most popular products in each category. The research paper will compare algorithms KNN, Decision tree based on their recommendation relevance, computational efficiency. This paper provides insights into the strengths and weaknesses of KNN and decision trees in collaborative filtering for e-marketing systems by comparing the two approaches. The results also provide a practical model for applying machine learning methods to enhance user's satisfaction.

2. Literature Review

One of the best-known methodologies of recommender systems are collaborative-filtering systems which try to determine user preferences based on what other users desire. The basic idea is that users who behave prior similarly, i.e. who have purchased similar items, rated items to give them similar ratings or used similar content, are more likely to exhibit similar taste in the future [7]. Algorithms which are applicable include K-Nearest Neighbors (KNN) and decision trees. KNN attempts to join users with similar taste and generate recommendations. Decision trees attempt to derive relationships based on user interaction to generate recommendations which might interest the user, but both enhance the personalization of recommendations given to the user [8]. These systems work best where there is an abundance of data, such as in ecommerce type sites, and enhance the user's ability to find products which formerly were not easily found through conventional search methods, enhancing satisfaction and user interest [9]. Collaborative filtering offers businesses enhanced user interactivity and increases sales with personalized recommendations based on user habits. Its versatility has confirmed its place in the pantheon of contemporary recommender systems in a wide variety of industries [10].

Kaya et al. (2022) proposed a hybrid research paper recommendation system based on deep learning, document similarity, clustering, and keypad extractions via a hybrid system reaching 80% accuracy and speedy admission of a useful item [11].

In 2023, Lukile et al. presented a personalized e-commerce recommendation system based on association rules using a recursive growth algorithm. The system enhanced customer engagement and the likelihood of purchasing products by making product suggestions highly relevant [12].

In 2020, Elshenawy et al. suggested an adaptive recommendation system with the application of data mining and multi-label classification to forecast the student performance and suggest appropriate engineering paths at Al-Azhar University [13].

Kongpeng et al. (2024) evaluated Decision Tree, Random Forest, KNN, and MLP algorithms to give recommendations on tourist destinations in Thailand with an average hit of 0.8 and NDCG of 0.59 which leads to the conclusion that the feature and algorithm selection can influence accuracy [14].

Ferdows (2024) also presented a hybrid movie recommendation system with text to numbers, cosine similarity and ALS algorithms for timeliness on the TMDB 5000 dataset, giving better accuracy in predicting the selection of movies made by users [15].

3. Methodology

This research paper uses two machine learning algorithms, the k-nearest neighbor (KNN) algorithm and the decision tree (DT) classifier, to create a collaborative filtering system for e-marketing products. The proposed research design will empirically model user purchasing behavior, using machine learning algorithms to provide personalized product recommendations. To achieve this, the process is divided into five main steps: data preparation, user-product matrix construction, algorithm implementation, recommendation generation, and purchasing behavior visualization, as in Fig.1.

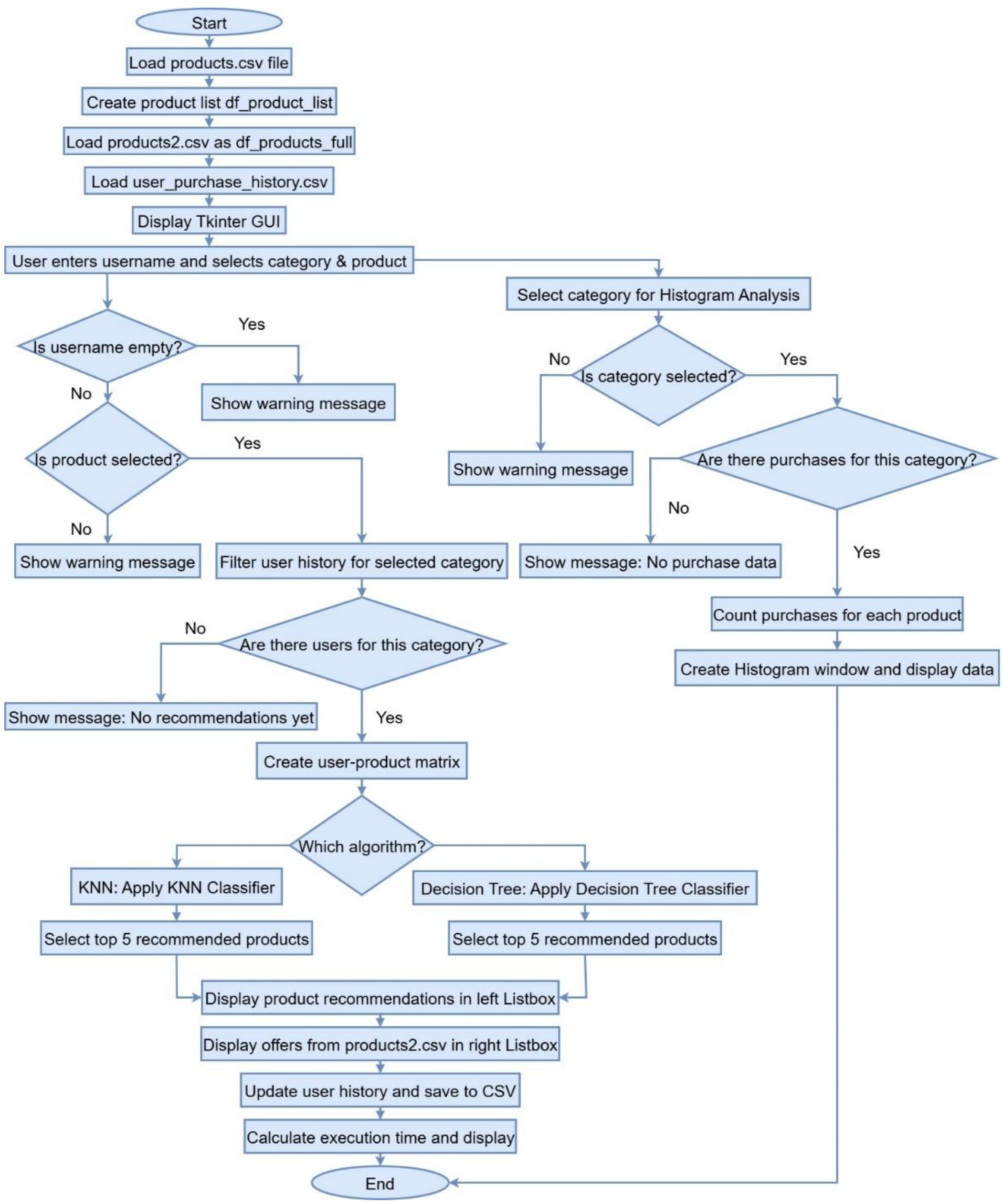

Fig. 1 - Flowchart of Product Collaborative Filtering (KNN vs Decision Tree).

The operational schematic of the solution may be captured as below:

1. Load dataset and pre-process product details.

2. Read/get user purchase history.

3. Create user–product matrix of the selected category.

4. Implement required algorithm (KNN or DT) to receive recommendations.

5. Show Top 5 recommendations with additional offers.

6. Update purchase history file dynamically.

7. Plot purchase frequency distributions based on histogram.

Each step in the flowchart is very important in providing sound recommendations. Data preparation provides clarity, consistency, and the proper organization of product information in preparation for evaluation. Constructing the user-product interaction matrix permits the system to remember each individual user's history of purchases, which is the base for the K-nearest neighbors system (KNN), and the decision tree algorithms. The updating of information can be done using algorithms which predict either based on user similarities (KNN), or learned decision rules (decision tree) along with this proceeds the dynamic updating of purchase history, to keep recommendations flexible and fluid. Finally, the frequencies of purchase are distributed through the use of histogram.

3.1. Data Preparation

The preprocessing phases consist of eliminating missing values, standardization of product names, flattening the product files and building the product look up dictionaries. User purchase history is used so updates may be made in real-time. Three data sets are used in this paper:

1. Prime Product Data Set (products.csv): which contains five categories (Food, Dairy, Canned Goods, Juices, and Detergents), with ten products in each.

2. Secondary Product Data Set (products2.csv): which contains the same categories but only four products per category, to be used as additional file of Offers.

3. User History of Purchases (user_purchase_history.csv): which contains a file to be updated dynamically with every purchase, storing user names, categories and products purchased.

3.2. User–Product Matrix

To accurately monitor user purchasing behavior and provide sound recommendations, an organized representation of the interaction between users and products is needed. This representation is provided in the user-product matrix which encodes a specific purchase made by a specific user of a product. This hierarchical manner in which the interaction is represented facilitates the application of machine learning algorithms for finding similarities in users and thereby providing valid suggestions. Each product category is represented as a binary user-product matrix. User are represented as rows, products as columns, and the values are allocated in the matrix as 1 if the product was purchased and 0 if it was not purchased.

$$Matrix\ size = (n\_users \times n\_products) \tag{1}$$

Where: n_users: total number of users in the system, n_products: total number of products in the selected category. This matrix serves as the input to both algorithms.

3.3. K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm is a collaborative filtering algorithm, which suggests products to a user, through the purchasing behavior of other users. A purchase vector represents each user, such that each value represents whether or not the user has purchased a particular item (1 or 0). The users are compared to the nearest neighbors thereby creating the similarity, and the comparison is made to come up with the closest neighbors whose preferences are then applied in explaining the personalized product recommendations. Euclidean Distance is used to measure the similarity of two user's u and v:

$$d(u, v) = \sqrt{\sum_{i=1}^{n} (x_{ui} - x_{vi})^2} \tag{2}$$

the symbols are defined as follows:

(d (u, v)): The Euclidean distance between users (u) and (v).

(u, v): the two users whose similarity is being measured.

(n): the total number of products in the selected category.

(x_ {u ,i }): the purchase status of product (i) for user (u) (1 if purchased, 0 if not).

(x_ {v ,i }): the purchase status of product (i) for user (v) (1 if purchased, 0 if not).

The nearest neighbors are those with the smallest distances. The probability that a user $\boldsymbol{u}$ will purchase a product $\boldsymbol{p}$ is estimated as:

$$P(u,p) = \frac{\sum_{v \in N(u)} x_{vp}}{|N(u)|} \tag{3}$$

Where:

$P(u,p)$: the estimated probability that user uu will purchase product pp.

uu: the target user for whom the recommendation is being made.

pp: the product being evaluated for recommendation.

$N(u)$: the set of kk nearest neighbors of user uu.

$x_{v,p}$: purchase status of product pp by neighbor vv (1 if purchased, 0 if not).

| $N(u)$ |: the total number of neighbors in the set $N(u)$N(u).

The products are then ranked by their estimated probabilities, and the top items are selected as personalized recommendations for the target user.

3.4. Decision Tree Classifier (DT)

Decision Tree (DT) method looks at each product as a binary classification problem. There is a decision tree that is trained with the purchase status of the other products in the same category forming the features and the target variable being whether to purchase the product or not (1) being the target variable. In the process of passing the purchase vector of a user through the trained tree, the likelihood of the user purchasing the product is obtained by the distribution of the samples in the leaf node that the user arrives at:

$$P(u,p) = \frac{\text{purchases of product } p}{\text{total samples in leaf}} \tag{4}$$

$P(u,p)$: the estimated probability that user uu will purchase product pp.

u: the target user for whom the recommendation is made.

p: the product being evaluated.

Leaf node: the terminal node of the decision tree where the user's vector ends up.

Numerator: number of users in the leaf who purchased product pp.

Denominator: total number of users in the leaf.

Products are then ranked based on their probabilities, and the top are recommended.

3.5. Recommendation Generation

The recommendation generation step combines the predictions of the selected algorithm (KNN or decision tree) to provide actionable recommendations to the user. The system provides two main outputs: the top five personalized product recommendations and additional offers based on a set of additional product data. The system is dynamic and can respond to changing user preferences by updating their purchase history after each execution, making future recommendations relevant and personalized.

3.6. Visualization of Purchase Patterns

The frequencies of purchases are given in histogram so as to bring out the popularity of products under each category. Arabic reshaping and bi-directional text control is used to enable the appropriate display to support a multilingual format.

Such visualizations do not only give a clear picture on the demand of the products, but also assist in the analysis of user behavior pattern. With trends and preferences discovered, E-Marketing Systems managers are able to make viable decisions regarding promotions, and enhancing appropriate relevance of recommendations, as in Fig.2.

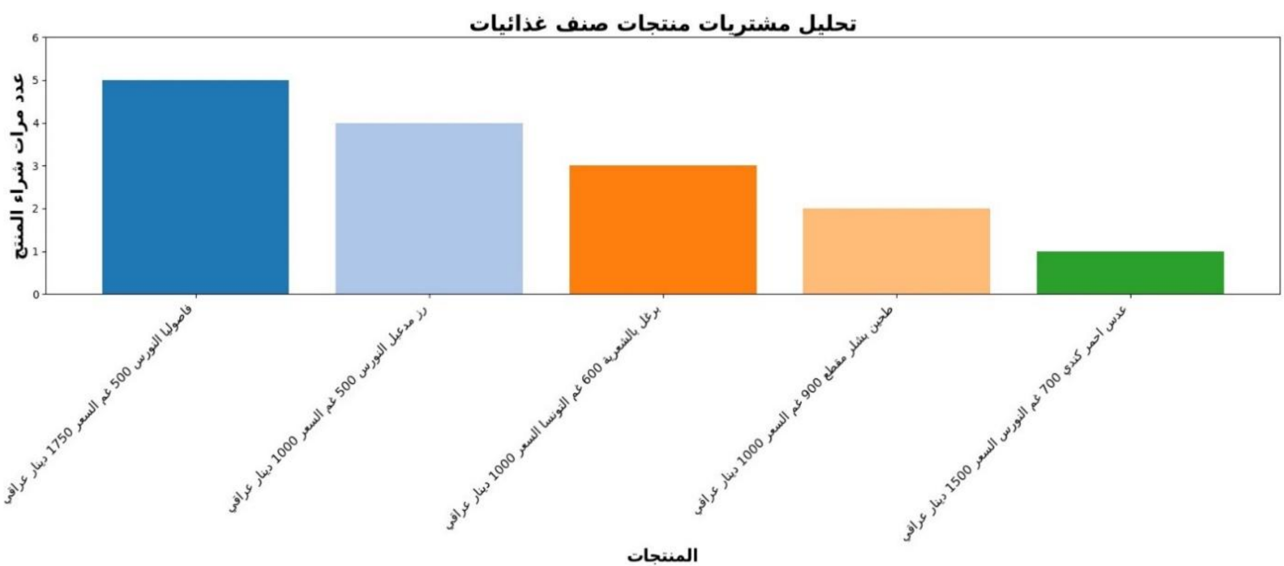

Fig. 2 -Histogram Analysis.

3.7. Results and Analysis

The performance results of the algorithms highlighted clear differences in the ability of the two algorithms to make recommendations. The KNN algorithm proved to be both more accurate and more reliable than the decision tree algorithm. Other metrics providing advantages to KNN in addition to accuracy include precision, recall, F1 -score and execution time. Unlike decision tree, which interprets rules of the decision and provides clear insights, decision tree performs poorly in the ability to make accurate recommendations, as shown in Table 1.

The KNN algorithm outperformed the Decision Tree algorithm in all the evaluation measures. The KNN had an accuracy of 0.9651 while the Decision Tree had an accuracy of 0.8392 which means that the recommendations made by the KNN are more valid. The precision of the KNN was 0.8233 which is 0.2094 the Decision Tree, which shows that the recommendation of the KNN algorithm are more specific. The recall for the KNN algorithm was equal to 0.8430, that is, the KNN algorithm identified the majority of products being purchased from users in the previous time period, whilst the Decision Tree had a corresponding recall of 0.2111 which is very low showing that a lot of data was missed. The F1 measure which measures the quality or power of the algorithms to recommend products confirms the KNN superiority, for the KNN got the high F1 score of 0.8293 whereas the Decision Tree netted the very low F1 score of 0.2072. The KNN seems to be the clear winner as can be seen in this case since it made very accurate and very much balanced recommendations and hence it can be also efficiently incorporated in the practical aspect of the business, whereas, that the Decision Tree on the other hand left something to be desired both in ways of efficiency as well as that of obtained recommendation power.

Table 1 - Performance Comparison: KNN vs Decision Tree.

Metric	KNN	Decision Tree
Accuracy	0.9651	0.8392
Precision	0.8233	0.2094
Recall	0.8430	0.2111
F1-score	0.8293	0.2072
Time (s)	0.896	3.686

The KNN algorithm outperformed the Decision Tree algorithm in all the evaluation measures. The KNN had an accuracy of 0.9651 while the Decision Tree had an accuracy of 0.8392 which means that the recommendations made by the KNN are more valid. The precision of the KNN was 0.8233 which is 0.2094 the Decision Tree, which shows that the recommendation of the KNN algorithm are more specific. The recall for the KNN algorithm was equal to 0.8430, that is, the KNN algorithm identified the majority of products being purchased from users in the previous time period, whilst the Decision Tree had a corresponding recall of 0.2111 which is very low showing that a lot of data was missed. The F1 measure which measures the quality or power of the algorithms to recommend products confirms the KNN superiority, for the KNN got the high F1 score of 0.8293 whereas the Decision Tree netted the very low F1 score of 0.2072. The KNN seems to be the clear winner as can be seen in this case since it made very accurate and very much balanced recommendations and hence it can be also efficiently incorporated in the practical aspect of the business, whereas, that the Decision Tree on the other hand left something to be desired both in ways of efficiency as well as that of obtained recommendation power.

The histogram shows the effect of the number of user histories on the performance of the two algorithms, as measured by key metrics as shown in Fig.3. The more information is found in the user histories, the greater the accuracy of KNN, which results in more accurate and more consistent predictions obtained. In contrast, the accuracy of the decision tree shows a slightly lower level of accuracy and only a low improvement in accuracy as the data grow, illustrating that it is less adaptable in this case.

It is evident that the accuracy of KNN increases vastly as more and more user data are acquired, which results in a greater number of accurate recommendations. The measurement of the decision tree, on the other hand, is now seen to be approximately constant at low values, signifying that its competence to filter out important products is of low value.

KNN also obtains a noticeably greater recall, since it has longer sample array in order to ascertain which products are the most relevant. The decision tree, however, shows only a slight improvement in comparison, often, as seems to be suggested, missing or obliterating similar products.

The KNN F1 score is representative of a continuous increase, which results in an equal degree of improvement in both precision and recall. The decision tree, however, remains at a low level, showing the overall low quality of its predictions. Both algorithms show that they slow in speed when the data in the array increase but show that KNN at last increases directly proportional, while the decision tree due to its tree structure increases very much faster in view of its complexity, the computational efficiency in the case of KNN being greater.

Overall the results (a-e) show that the historical data supplied by the users has a great and important effect on improving the performance of KNN, irrespective of the test used, but that decision trees in these cases seem slow in response and sensitive to the effects of change.

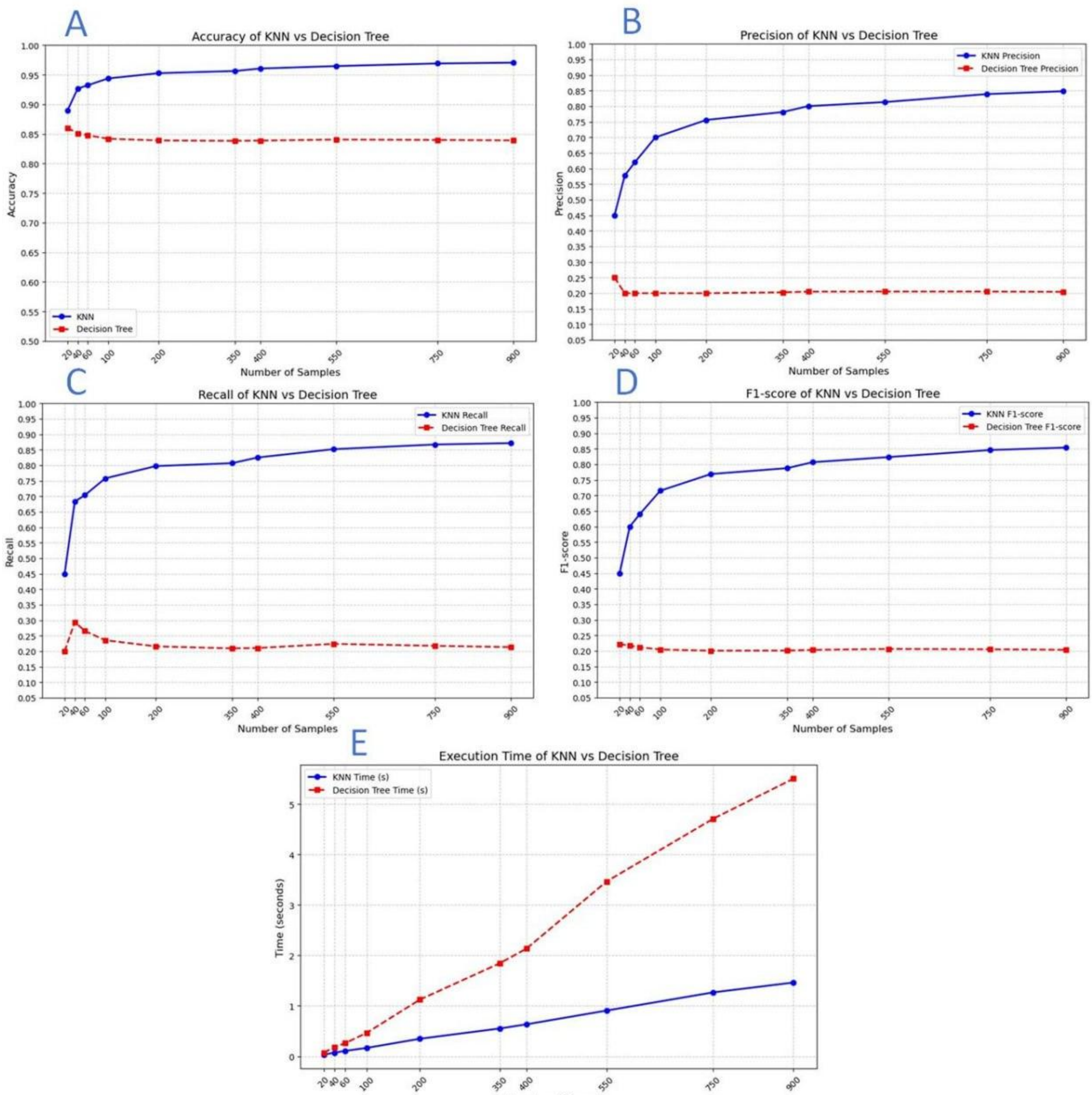

Fig. 3- METRICS A: Accuracy, B: Precision, C: Recall, D: F1_score, E: Execution Time.

4. Conclusion

This research paper presents collaborative filtering of supermarket products using k-nearest neighbor (KNN) and decision tree (DT) algorithms. Using the product data files and the user log file, the system was able to generate personalized recommendations of the top five products, as well as the number of times users had acquired the products, using the histogram. The results showed that KNN outperformed the DT algorithm in terms of accuracy, precision, recall and F1-score, thus being very effective in generating product recommendations. The results of the benchmark tests showed great differences of both algorithms, in terms of: accuracy (0.9651 for the KNN algorithm, and 0.8392 for the DT algorithm), precision (0.8233 for the KNN algorithm, and 0.2094 for the DTC algorithm), recall (0.8430 for the KNN algorithm, and 0.2111 DTC), F1-score (0.8293 for the KNN algorithm and 0.2072 for the DTC algorithm). The performance measures for the KNN algorithm continued to improve as the data size increases, in terms of: precision, recall and F1-score. This indicates that KNN uses more data out of the user's history and thus obtains better recommendations of the products. On the other hand, decision tree shows only a slight improvement as the data size undergoes an upward change, thus being less effective to the ability of adapting to larger data. The time taken for computations of the KNN algorithms was less in comparison to that of the decision tree algorithm. The KNN algorithm is able to compute data more effectively due to their simpler data structure and operations, and executes the correct results, where performance is concerned, especially in the case of both small and medium data size. With increase in data size station time of operations get slowed down for both algorithms while KNN still is able to outperform the results for the decision tree algorithm.

References

[1] H. Khatter, S. Arif, U. Singh, S. Mathur, and S. Jain, "Product recommendation system for E-commerce using collaborative filtering and textual clustering," in 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 612–618.

[2] N. B. Munkholm, R. Alphinas, and T. Tambo, "Collaborative filtering, K-nearest neighbor and cosine similarity in home decor recommender systems," arXiv Prepr. arXiv2402.06233, 2024.

[3] D. Prasad, S. K. Goyal, A. Sharma, A. Bindal, and V. S. Kushwah, "System model for prediction analytics using k-nearest neighbors algorithm," J. Comput. Theor. Nanosci., vol. 16, no. 10, pp. 4425–4430, 2019.

[4] D. A. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method," Appl. Comput. Informatics, vol. 12, no. 1, pp. 90–108, 2016.

[5] M. S. Abd Rahman, N. A. A. Jamaludin, Z. Zainol, and T. M. T. Sembok, "The application of decision tree classification algorithm on decision-making for upstream business," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 8, 2023.

[6] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: from efficient prediction to responsible AI," Front. Artif. Intell., vol. 6, p. 1124553, 2023.

[7] P. K. Singh, P. K. D. Pramanik, and P. Choudhury, "Collaborative filtering in recommender systems: Technicalities, challenges, applications, and research trends," in New Age Analytics, Apple Academic Press, 2020, pp. 183–215.

[8] G. Chornous, I. Nikolskyi, M. Wyszyński, G. Kharlamova, and P. Stolarczyk, "A hybrid user-item-based collaborative filtering model for e-commerce recommendations," J. Int. Stud., vol. 14, no. 4, 2021.

[9] Z. R. Basystiuk and Z. Rybchak, "Recommendation systems in e-commerce applications," Inf. Syst. Networks, vol. 15, pp. 252–259, 2024.

[10] H. H. L. Nguyen, T. N. Le, T.-T. Ha, and L. V. Nguyen, "Collaborative Filtering-Based Personalized Recommendations: Challenges, Limitations, and Applications," in International Conference on Intelligence of Things, 2024, pp. 12–23.

[11] E. Gündo\ugan and M. Kaya, "A novel hybrid paper recommendation system using deep learning," Scientometrics, vol. 127, no. 7, pp. 3837–3855, 2022.

[12] M. Loukili, F. Messaoudi, and M. El Ghazi, "Machine learning based recommender system for e-commerce," IAES Int. J. Artif. Intell., vol. 12, no. 4, pp. 1803–1811, 2023.

[13] M. Ezz and A. Elshenawy, "Adaptive recommendation system using machine learning algorithms for predicting students best academic program," Educ. Inf. Technol., vol. 25, no. 4, pp. 2733–2746, 2020.

[14] S. Kongpeng and A. Hanskunatai, "Tourist Destination Recommendation System based on Machine Learning," in Proceedings of the 2024 9th International Conference on Big Data and Computing, 2024, pp. 58–67.

[15] M. D. R. Hasan and J. Ferdous, "Dominance of AI and machine learning techniques in hybrid movie recommendation system applying text-to-number conversion and cosine similarity approaches," J. Comput. Sci. Technol. Stud., vol. 6, no. 1, pp. 94–102, 2024.