Adv Mobile Learn Educ Res, 2025, 5(1): 1388-1399
DOI: 10.25082/AMLER.2025.01.013

RESEARCH ARTICLE

Smartwatch Selection Recommendation System Using the K-Nearest Neighbor (KNN) Algorithm with Dynamic Dataset Optimization

Gino Erman Agusta[1*] **Fatchul Arifin**[1]

[1] Department of Electronics and Informatics Engineering Education, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia

Correspondence to: Gino Erman Agusta, Department of Electronics and Informatics Engineering Education, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia; Email: ginoerman.2024@student.uny.ac.id

Received: January 24, 2025;
Accepted: April 20, 2025;
Published: April 25, 2025.

Citation: Agusta, G. E., & Arifin, F. (2025). Smartwatch Selection Recommendation System Using the K-Nearest Neighbor (KNN) Algorithm with Dynamic Dataset Optimization. *Advances in Mobile Learning Educational Research*, 5(1), 1388-1399. https://doi.org/10.25082/AMLER.2025.01.013

Abstract: This research aimed to develop a smartwatch recommendation system using the K-Nearest Neighbor (KNN) algorithm with dynamic dataset optimization. By employing a dynamic dataset, the accuracy of KNN calculations was enhanced. The dataset, stored in CSV format, was filtered based on user preferences when searching for a smartwatch, generating a dynamic dataset tailored to individual needs. The research involved 35 respondents to evaluate the precision and feasibility of the application. Results showed that 25.7% of respondents found the application highly relevant to their preferences, 31.4% relevant, and 31.4% somewhat relevant. User satisfaction levels indicated that 34.3% were very satisfied, 34.3% satisfied, and 20% somewhat satisfied, highlighting the application's effectiveness in meeting user expectations.

Keywords: dynamic dataset, K-Nearest Neighbor, recommendation system, smartwatch

1 Introduction

In the increasingly advanced digital era, smart devices such as smartwatches have become devices that are increasingly in demand by various groups. Its diverse functionality, from health activity tracking, and smart notifications, to productivity features, makes smartwatches an attractive choice in the era of increasingly connected technology (Chandel et al., 2022). However, a variety of smartwatch options in different markets in terms of brands, models, battery capacity, health features, device compatibility, and price, can make it difficult for consumers to determine the product that suits their needs. Consumers who do not understand the details of the specifications tend to take a lot of time to choose and often choose a device that is less than optimal for their needs (Wu et al., 2016). This condition indicates the need for a recommendation system that can help users choose a smartwatch based on their preferences and needs (Reeder & David, 2016). This problem is the main reason for the need for a reliable and effective recommendation system to help potential buyers determine the right smartwatch choice. This research aims to develop a recommendation system using the K-Nearest Neighbor (KNN) method, which can provide recommendations based on the proximity of user preferences to product data in the market. The KNN method was chosen because of its simplicity and effectiveness in processing big data and generating recommendations based on Euclidean distance calculations (Triguero et al., 2019). This system is expected to be a solution that makes it easier for consumers to choose a smartwatch without having to understand all the technical specifications in depth.

This KNN-based recommendation system will utilize dynamic datasets, namely datasets whose columns are constantly updated according to what is the user's preference in choosing a smartwatch (Liu et al., 2021). If you use all the existing preferences, then the data becomes less accurate. For example, users only want to search for smartwatches based on price and GPS features. The usual KNN algorithm will compare the 2 preferences that the user is searching for with the entire preference data (Patro et al., 2020). This causes the calculation of distances to be inaccurate because Euclidean distances are also sought even though such preferences are not needed. By using dynamic datasets, the user's preferences will only be compared to the 2 preferences in the database (Ebrahimi, 2024). This approach is expected to increase the accuracy and relevance of the recommendations provided by the system. In addition, the development of an intuitive and easy-to-use user interface is also an important factor in the success of this recommendation system (Murphy-Hill & Murphy, 2013). A simple but effective interface can make it easier for users to convey their preferences without having to understand complicated technicalities (Johnson, 2020). This is very important, considering that not all users have in-depth knowledge of the specifications of the smartwatch, so the recommendation system must be able to adapt to different levels of understanding and user needs.

A recommendation system is a technology that aims to assist users in choosing a product or service that suits their preferences or needs (Konstan & Riedl, 2012). By utilizing existing user data and information, the system can provide suggestions or options based on behavior patterns, preferences, and other relevant data. Various methods have been used in the development of recommendation systems, such as content-based methods, collaborative filtering, and hybrid-based methods (Ibrahim et al., 2023). In the context of smartwatch purchases, the recommendation system is designed to provide recommendations based on user specifications, features, and preferences (Ray & Singh, 2025). Smartwatches are wearable devices that have various functions such as health monitoring, message notifications, and additional features to support daily activities (King & Sarrafzadeh, 2018). The main criteria for choosing a smartwatch often include technical specifications such as battery capacity, operating system compatibility, screen size, health features, and price (Ramezani et al., 2023). With more and more options on the market, consumers need an effective recommendation system to help them choose a smartwatch according to their needs.

K-Nearest Neighbor (KNN) is an algorithm that is popularly used in data classification and prediction (regression) based on the principle of data proximity (Halder et al., 2024). KNN falls into the category of instance-based algorithms, where these algorithms do not have a training phase traditionally (Song et al., 2017). KNN works by calculating the distance between the new data and the existing data and then taking the number of closest neighbors to determine the results of the recommendation or prediction (Adeniyi et al., 2016). In the KNN method, the distance between data points is measured using various methods, one of which is the Euclidean distance (Adeniyi et al., 2016). Euclidean distance measures the proximity between data based on the dimensions of its attributes, so the smaller the distance between two data, the more similar or relevant the data is to each other (Alfeilat et al., 2019).

The application of KNN in the recommendation system utilizes input data from user preferences to search for products that have the highest proximity based on the desired criteria (Johnson, 2020). In this case, the KNN method can generate personalized recommendations, where users get recommendations that suit their unique needs. This algorithm is also capable of handling large and dynamic data, making it suitable for use in diverse markets such as smartwatches (Nurwanto et al., 2016). One of the advantages of KNN is the simplicity of its implementation (Dhanabal & Chandramathi, 2011). This algorithm does not require many assumptions and is very flexible in handling data at various scales. However, the weakness of KNN lies in its sensitivity to irrelevant data or noise (Syriopoulos et al., 2022). Therefore, good data preprocessing is needed to improve the accuracy of the recommendation results. Data collection in the recommendation system is carried out by combining product specification information as well as user preferences. This data can be in the form of technical attributes, such as screen size, health features, and the price of the smartwatch. User preferences are derived from surveys or direct interactions that reflect their specific needs. A structured data collection process will improve the quality of recommendation results. Validation of the recommendation system is important to ensure that the algorithms used provide relevant results. Testing can be conducted by involving users who represent a specific market segment, where the results of the recommendations are measured by their level of satisfaction with the recommended product. In addition, validation methods such as cross-validation can also be used to measure the accuracy of the KNN algorithm in providing the right recommendations.

The increasing variety of smartwatches presents a challenge for consumers in selecting devices that best match their needs. Recommender systems address this issue by filtering choices based on user preferences. While existing studies have explored various techniques for smartwatch recommendations, they often rely on static datasets, limiting their adaptability to changing user preferences and market trends. This research aims to bridge this gap by introducing a dynamic dataset optimization mechanism within the KNN algorithm, enhancing its ability to deliver personalized and up-to-date recommendations.Several recommendation approaches have been employed for smartwatch selection, including collaborative filtering (CF), content-based filtering (CBF), and hybrid models. CF relies on user interactions, whereas CBF analyzes item attributes. While these methods offer varying degrees of accuracy, they often fail to adjust dynamically to new data. Recent advancements in recommender systems integrate machine learning techniques such as neural networks and deep learning; however, these methods require extensive computational resources. Our study builds upon existing KNN implementations but enhances them by incorporating real-time dataset updates, addressing the limitations of static dataset dependency.While KNN is widely used for recommendation tasks due to its simplicity and interpretability, its reliance on a static dataset limits its adaptability. Existing research has not adequately explored the impact of dynamic dataset optimization in KNN-based recommendation systems, particularly in the context of smartwatch selection. This study fills this gap by introducing a novel mechanism that updates the dataset based on real-time market trends and evolving user preferences. Our contributions include: a dynamic dataset optimization approach that enhances KNN's adaptability, a comparative performance analysis against traditional KNN and other recommendation methods, and demonstration of improved recommendation accuracy and relevance through experimental evaluations

This research contributes by presenting a recommendation system developed with the KNN algorithm based on the price and features of smartwatches that are optimized by using dynamic datasets according to preferences. Thus, the accuracy and relevance of recommendations can be improved. This research also builds a foundation for the enrichment of the use of the KNN method in the context of dynamic and personalized consumer technology product recommendations. In addition, the research focuses on developing an interface that makes it easy for users to convey their preferences simply. With this approach, the recommendation system can be accessed and used by various circles without significant technical barriers. Therefore, this research not only provides practical solutions in the selection of smartwatches but also enriches theoretical studies on the implementation of the KNN algorithm in the recommendation system. Furthermore, the research also opens up opportunities for further exploration in the development of a recommendation system with the KNN algorithm capable of considering more parameters, such as user reviews, market trends, and product availability. By expanding the scope of data processed by the recommendation system, consumers will not only get more accurate recommendations but also more relevant to their evolving needs.

2 Materials and methods

This research used a quantitative approach to develop and evaluate a smartwatch purchase recommendation system based on the K-Nearest Neighbor (KNN) algorithm. This research aimed to validate the effectiveness of KNN in providing recommendations that were in line with user preferences. The data used included smartwatch specifications as well as user preferences and needs obtained through surveys. This research was carried out in several stages, namely (1) data collection, (2) data preprocessing, (3) development of a KNN-based recommendation system, (4) system testing, and (5) evaluation of recommendation results (Sharma, 2024). In the context of a smartwatch recommendation system, the Euclidean distance is used to measure the conformity between user preferences and existing product specifications. Euclidean distance calculation formula (Alfeilat et al., 2019):

$$dist\ (x_1, x_2) = \sqrt{\sum_{i=0}^{n} (x_{1_i} - x_{2i})^2} \tag{1}$$

Accuracy measurement used a satisfaction survey whose results were calculated using precision metrics, where the correct number of recommendations was calculated against the total recommendations produced. The formula for calculating precision metrics was as follows:

$$\text{Precision} = \frac{\text{Relevant Recommendations Generated}}{\text{Total number of Recommendations Generated}} \tag{2}$$

2.1 Data Collection

The dataset contains 70 smartwatch records sourced from official product websites and e-commerce platforms, including features such as Bluetooth Call, NFC, GPS, WiFi, Battery Capacity, and Health Features. To address potential biases, data were collected from multiple brands across different price ranges. Additionally, feature selection was performed to eliminate redundant attributes, ensuring the dataset represents diverse smartwatch capabilities. The data were taken from the official product website info and/or product detail information on e-commerce sold by the official shop. The data taken were Price, Brand, Model, and Image link. As for the features that were owned by the smartwatch, the data needed were Bluetooth Call, NFC, GPS, WiFi, Temperature Sensor, SIM card, Voice Command/AI, Waterproof, Battery (mAh), Standby Time, Screen Type, Screen Size (inch), Camera Control, Music Control, Health Features, Sport Assistance (Modes), and Bluetooth Version. The data were stored in a CSV file as a smartwatch database, as shown in Table 1.

Table 1 Dataset Saved in CSV File

Brand	Price	Model	...	Rescale_SA	BT_Vers
RAYVASI	0.011570	GT PRO	...	1.0	5
AOLON	0.026033	GT5 PRO	...	1.0	5.1
HAYLOU	0.029732	RS5 LS19	...	1.0	5.3
COROS	0.865532	VERTIX 2S	...	0.0	5.3
SUUNTO	0.414839	RACE S	...	1.0	5.0
AMAZFIT	0.057581	Bip 5	...	1.0	5.2
COLMI	0.007803	C8 Max	...	1.0	5.3
GOJODOQ	0.003767	CURVE FB032	...	1.0	NaN
MIBRO	0.074062	GS PRO	...	1.0	5.3
SUPERCALLA	0.038813	DM56	...	1.0	5.3
SAMSUNG	0.293085	Galaxy Watch 7	...	0.5	5.3
APPLE	0.495493	Watch Series 9	...	0.0	5.3
XIAOMI	0.038073	Watch 5 Lite	...	1.0	5.3
SKMEI	0.014193	S233	...	1.0	5.0
OLIKE	0.011032	Zenith W15	...	0.0	5.2
APPLE	1.000000	Watch Ultra 2	...	0.0	5.3
HAYLOU	0.014933	Solar Neo	...	1.0	5.3
COROS	0.286964	Pace 3	...	0.0	5.3
SAMSUNG	0.051527	Galaxy Fit 3		0.5	5.0
XIAOMI	0.179335	Watch 2 pro		1.0	5.2
AMAZFIT	0.197498	Cheetah		1.0	5.3
ITEL	0.006458	ISW-O23 Active		0.0	5.3
RAYVASI	0.016951	COMPASS		1.0	5.0
AOLON	0.077425	NAVI R4		1.0	5.3
OLIKE	0.013386	Lumi R1 Curved		0.0	5.3
ITEL	0.007803	ISW-O43	...	0.0	5.3
...	...	...	...	...	...
HUAWEI	0.052200	Watch Fit Se	...	1.0	5.0

2.2 Data Preprocessing

Preprocessing involved:

(1) Handling Missing Values: Missing numerical data was imputed using the median, while categorical data were assigned the mode value.

(2) Outlier Detection: Price and battery capacity outliers were detected using the IQR method and treated appropriately.

(3) Feature Scaling: Min-max normalization was applied to numerical features to standardize feature weights, avoiding KNN bias towards large-scale attributes.

(4) Categorical Encoding: Nominal variables such as brand popularity and screen type were encoded using binary and ordinal encoding techniques.

2.3 Cleansing Data

The data obtained will be checked again so that there are no errors in the next process (Abedjan et al., 2016). The cleansing process includes: (1) Removing Duplicates: Deleting data that has the same specifications or information so as not to affect the results of the analysis (Ganti & Sarma, 2013); (2) Filling in the Data Blank (Missing Value): If there is blank data, imputation is carried out with methods such as average, median, or default value, depending on the type of data (Sulistyo et al., 2020); (3) Format Normalization: Align data formats, such as units of measurement (inches, mAh), to be uniform and consistent (Bharadwaj et al., 2024); (4) Data Validation: Ensuring that the data used is correct, for example, double-checking prices or features based on official sources (Mertz, 2021).

2.4 Data Transformation

Reference data must be converted into numbers to be calculated using the KNN (K-Nearest Neighbors) algorithm (Zhang et al., 2017). Of all the existing specifications, 13 columns are used that will be KNN references. The transformation process includes: Data grouping and encoding: Some inputs have been converted into a language that is easy for users to understand, such as:

(1) Brands: famous-unfamous, this is based on an initial survey, of users' knowledge of the brands in the data. Famous is assigned to 1, not famous 0.

(2) Screen size: wide-medium-small. This division is based on all the screen sizes included in

the data, and then the data is divided into 3 using the equal-length interval method. So the small screen range (< 1.3067) is obtained with a value of 0, the medium screen (1.3067 – 1.6634) is assigned to 0.5 and the widescreen (> 1.6634) is assigned to 1.

(3) Resistance in water: non-resistant assigned to 0, splash resistant assigned to 0.5, and water resistance assigned to 1.

(4) Screen type: good (Amoled, Super Amoled, IPS, Retina) is asign to 1 and regular is asign to 0.

(5) Battery life: non-durable (1 – 7 days) is asign to 0, moderately durable (7 to 21 days) is asign to 0.5, and durable (> 21 days) is asign to 1.

(6) Several sports mode features: few (< 31) is assigned to 0, medium (31 – 100) is assigned to 0.5, and many (> 100) are assigned to 1.

Data with two categories will be changed to 0 and 1 such as GPS, NFC, Bluetooth call, Wifi, sim card, temperature sensor, Voice command, camera control, and music control. Value Normalization: Numerical values, such as prices, are normalized using the min-max scaling normalization technique to make the weights between features less uneven.

2.5 Frontend and Backend Design

For the application interface, the frontend design is created using: (1) CSS Framework Bootstrap: Which provides a responsive and modern look to the application, with ready-to-use components such as a grid system, cards, buttons, and input forms (Laaziri et al., 2019). (2) JavaScript and Ajax: Used to add interactivity to the page, and asynchronous communication between the frontend and backend, so that data can be displayed or updated without the need to reload the page (Usmani, 2023). (3) Python: as a backend, to process data using the KNN algorithm by importing the library as follows (Wang et al., 2023):

(1) Flask: to render HTML pages;
(2) Pandas: Reading CSV files;
(3) Numpy: Mathematical operations;
(4) Scikit-learn: KNN algorithm and min-max normalization.

2.6 System Work Design

Flowcharts will be used to explain the workflow of the recommendation system using the K-Nearest Neighbor (KNN) method (Maillo et al., 2017). Figure 1 shows the design of the data process architecture with K-Nearest Neighbor. Users can search for smartwatch recommendations by filling in the criteria for the desired smartwatch. Filling in the criteria is adjusted to the features that the user wants. Users must enter the price with the features that are cool to the user. The choice taken by the user will be calculated using the calculation of the K-NN method, namely the Euclidean Distance formula. Then the distance calculation is sorted by the nearest distance. The nearest distance will be taken to be used as a recommendation to the user. So that the final result is a smartwatch recommendation based on user criteria.

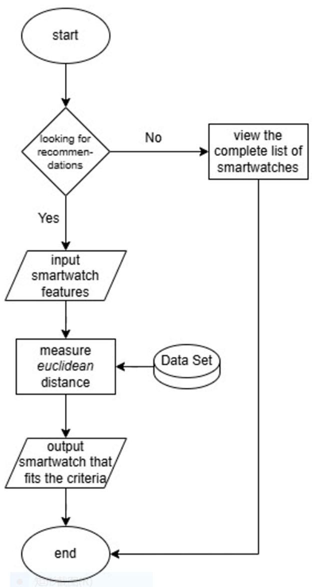

Figure 1 Data Process Architecture Design with *KNN*

Figure 2 is a use case diagram of a user showing the interactions that can be done by users on the website.

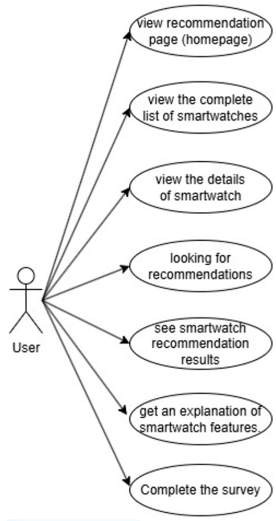

Figure 2 Use Case Diagram User

Users can register, view the home/page to find smartwatch recommendations, see all available smartwatch lists, see detailed smartwatch specifications, see an explanation of the features of the smartwatch, and fill out a survey. All done by the user without logging in first.

3 Results

3.1 User Preference-Based Smartwatch Recommendation

Here's a test of the system if the user is looking for a smartwatch that has a budget criterion of Rp. 500,000 medium screen size, good screen type, has Bluetooth call and voice assistant features, while other preferences are left by default (value 99), as depicted in Figure 3.

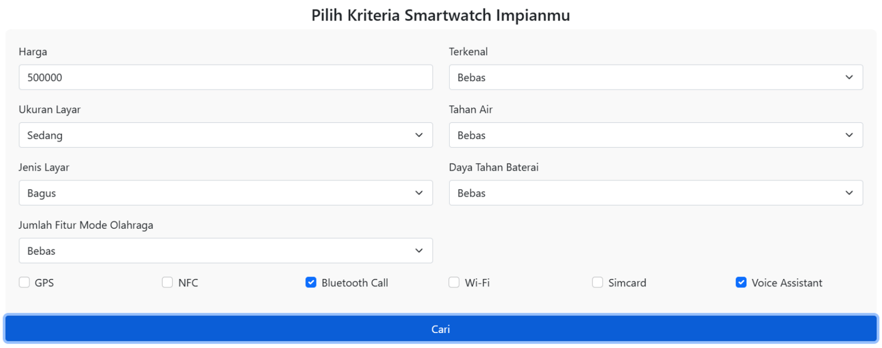

Figure 3 User's Preferred Display in Searching for Smartwatch Recommendations

The proposed smartwatch recommendation system employs a K-Nearest Neighbors (KNN) approach to match user preferences with the most suitable smartwatch options. Figure 3 illustrates a sample scenario where a user searches for a smartwatch with a budget of Rp. 500,000, a medium screen size, a high-quality screen type, and Bluetooth call and voice assistant features. After the user inputs these preferences and presses the "Search" button, the data is transmitted to the backend via Ajax. The dataset used in the KNN computation consists of 13 selected features out of 27 available columns, with price values scaled using the min-max normalization method, as displayed in Table 2.

3.2 Dataset Filtering and Dynamic Adjustment

To enhance recommendation accuracy, the system dynamically filters columns based on user input. Only columns corresponding to preferences with values other than the default (99) are considered, ensuring a more tailored recommendation process. Figure 4 displays the filtered dataset structure, while Table 3 presents an example of the adjusted dataset after column filtering.

Table 2 Default Dataset of 27 Columns

Price	Brand	Popular	...	Sport	Rescale_SA	BT_Vers
0.011570	RAYVASI	0	...	100	1.0	5
0.011166	RAYVASI	0	...	10	0.0	5
0.016951	RAYVASI	0	...	100	1.0	5
0.005180	RAYVASI	0	...	4	0.0	5
0.026033	AOLON	0	...	100	1.0	5.1
0.020651	AOLON	0	...	100	1.0	5.2
0.023947	AOLON	0	...	178	1.0	5.2
...	...	...	...	...	...	...
0.299072	HUAWEI	1	...	100	1.0	5.2

```
columns that used in KNN :
Index(['harga', 'rescale_screen', 'rescale_screenType', 'bt_call', 'ai'], dtype='object')
```

Figure 4 Print out the Column Names of Dynamic Datasets in Python Programming Language

Meanwhile, the dataset was still taken from the default dataset as a reference by only taking the required columns. The filtered dataset table was shown in Table 3.

Table 3 Dynamic Dataset After Column Filtering

Price	Rescale_Screen	Rescale_ScreenType	BT_Call	AI
0.011570	0.5	0	1	1
0.011166	1.0	0	1	0
0.016951	0.5	0	0	1
0.005180	0.0	0	1	1
0.026033	0.5	0	1	1
0.020651	1.0	0	1	1
0.023947	0.5	1	1	1
...	...	...	...	...
0.299072	0.5	1	1	0

3.3 Performance Evaluation Using Euclidean Distance

The distance between the user's input preferences and available smartwatches in the dataset is calculated using the Euclidean distance metric. Table 4 shows the results for K = 7, ranking smartwatches from the closest to the farthest based on their similarity to user preferences. The final recommendation list is determined by selecting the top K = 3 closest matches, which are then formatted in JSON and displayed on the user interface (Figure 5).

Table 4 Results of Distance Measurement Using KNN if K Equal to 7

Seq. Data	Price	Brand	....	Sport	Rescale_SA	BT_Vers	Distance
6	489.000	AOLON	....	178	1.0	5.2	0.000740
18	389.000	COLMI	....	100	1.0	5.0	0.007467
47	369.000	Itel	....	10	0.0	5.3	0.008812
8	1.284.000	AOLON	....	100	1.0	5.3	0.052738
25	1.999.000	Samsung	....	39	0.5	5.0	0.100834
35	799.000	XIAOMI	....	150	1.0	5.2	0.154648
30	4.490.000	Apple	...	30	0.0	5.0	0.268398

In the results of this calculation in Table 4, the order of data in the database was also obtained, which was then used to filter the database according to the data sequence number. The K value used in this case was 3, so 3 data were taken with the smallest distance. The data obtained were then converted into JSON format, which was then rendered on the user page (Figure 5).

3.4 Precision measurement and application feasibility

Based on the results of the survey that has been filled out by 35 respondents, show that the smartwatch purchase recommendation application has a fairly good level of precision. A total of 65.7% of respondents stated that the recommended smartwatch followed their budget, indicating that the application can provide recommendations relevant to the user's financial ability, making it easier for them to make a purchase decision.

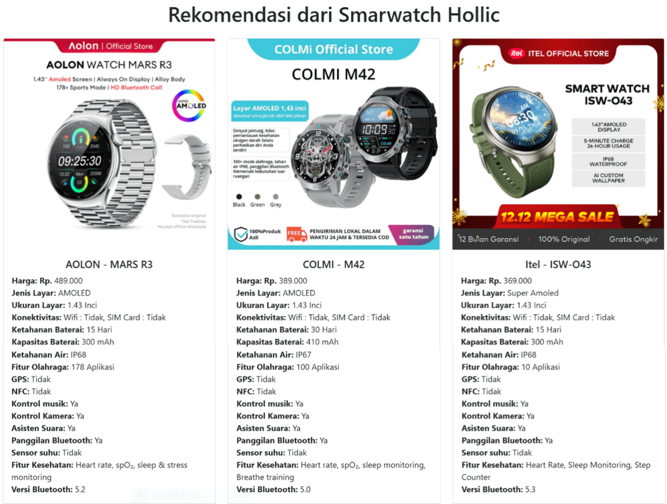

Figure 5 Display of Smartwatch Recommendation Results on The User Page

In addition, 48.6% of respondents stated that the main features of the recommended smartwatch met their needs. This indicates that the app has successfully identified the user's needs, although there are still opportunities to improve accuracy on certain features. The results of the survey on the relevance of recommendations to user preferences are shown in Figure 6.

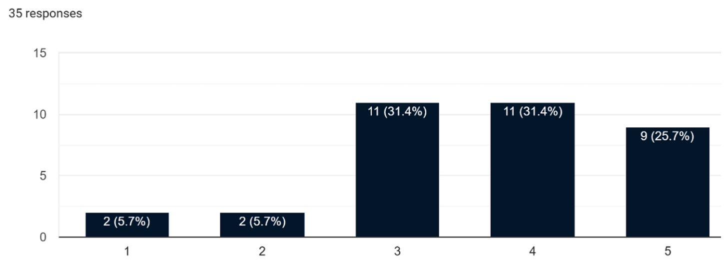

Figure 6 Survey Results of The Relevance of Recommendations to User Preferences

Figure 6 shows the assessment of the relevance of recommendations using the Likert scale (1-5), as many as 25.7% of respondents gave a score of 5 (very relevant), 31.4% gave a score of 4 (relevant), and 31.4% gave a score of 3 (quite relevant). With a total of 88.5% of respondents giving a score of 3 or above, this result shows that the majority of users feel that the app recommendations are relevant to their needs. As for the results of the survey on the level of user satisfaction with the application, it is shown in Figure 7.

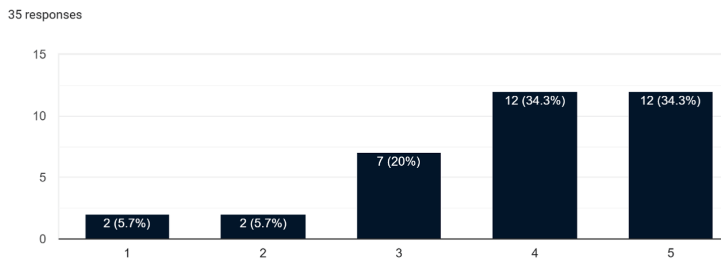

Figure 7 Survey Results of User Satisfaction With the Application

The results of the user satisfaction survey on the application in Figure 7 show that the level of user satisfaction with the application is also quite high. Based on the Likert scale (1-5), 34.3% of respondents gave a score of 5 (very satisfied), 34.3% gave a score of 4 (satisfied), and 20% gave a score of 3 (quite satisfied), with a total of 88.6% of respondents feeling quite satisfied to very satisfied. This high satisfaction rate indicates that the application provides an adequate experience in helping users choose a smartwatch. In addition, as many as 74.3% of respondents stated that they were willing to recommend this application to others. This reflects the user's trust in the app, which is considered useful and worthy of being used by others. Meanwhile, only 20% of respondents said they would not recommend the app, and another 5.8% said it may or depends on the situation, providing an opportunity for further improvement to improve overall user trust.

3.5 Critical Analysis and Comparison with Existing Smartwatch Recommendation Systems

One major limitation of the current study is the lack of a direct comparison with existing smartwatch recommendation systems. While the survey-based evaluation indicates moderate success, it does not conclusively establish whether the proposed approach performs significantly better than traditional recommendation systems employed by e-commerce platforms. To strengthen the evaluation, a comparative study using benchmark datasets from existing recommendation systems should be incorporated. Additionally, performance metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) should be calculated to quantify the precision of the KNN-based recommendations against alternative machine learning models, such as collaborative filtering or deep learning-based recommendation engines.

3.6 Evaluating Algorithmic Limitations and Potential Improvements

The discussion section previously attributed user dissatisfaction primarily to budget constraints. However, other algorithmic factors might have contributed to discrepancies in recommendation accuracy. Potential issues include: (1) Feature Weighting: The current system treats all features equally, whereas some features (*e.g.*, budget) may have a stronger influence on user decisions than others (*e.g.*, brand popularity). Implementing a weighted KNN approach could improve accuracy; (2) Cold Start Problem: The system does not adequately address cases where a user has minimal prior interactions. Integrating collaborative filtering techniques could mitigate this limitation; (3) Scalability: The KNN approach may not perform efficiently as the dataset grows. Exploring alternative algorithms, such as Approximate Nearest Neighbors (ANN) or matrix factorization, could enhance computational efficiency.

4 Discussion

The survey results show that there are 36.7% of respondents stated that application recommendations are not by their preferences. After further exploration, as many as 6 out of 12 respondents in this group chose smartwatches with quite a lot of features but had a budget of less than Rp. 500,000. This shows that some users have high expectations for features, but are limited by a very small budget. It is evident from the question about the reason for not buying a smartwatch, that 8 respondents gave reasons because of budget problems.

This phenomenon indicates a gap between user expectations and market realities, where smartwatches with many features generally have a higher price. This situation can be caused by a lack of user education about the limitations of features on smartwatches in a certain price range or a lack of adjustment to the application algorithm in recommending more realistic products according to the budget. To improve the satisfaction of this group of users, apps may consider displaying recommendations that provide clearer information about the limitations of features within a given budget, while also providing other options that can be compromised, such as smartwatches with essential features that remain relevant to their primary needs. In addition, a "buying guide" feature that educates users about the relationship between price and features can help them make more rational and realistic decisions. This will not only increase user satisfaction but also reinforce the perception that the app can understand the needs and preferences of users more deeply. This phenomenon also opens up opportunities for further development in recommendation algorithms. For example, the algorithm can be updated to take into account not only the user's feature preferences but also their budget constraints as key parameters. Thus, recommendations can be designed more adaptive and realistic, featuring products that suit the user's financial condition without overriding their primary needs.

Additionally, the algorithm can be equipped with a feature of grouping products based on a specific price range, so users can easily explore relevant options within their budget. Related research shows that personalizing recommendations based on budget and user preferences can improve relevance and user satisfaction. For example, research by Oktaviani et al. (2024) found that a recommendation system that takes into account the user's budget limitations in e-commerce increases conversion rates. Similar results were also found in research by Zikry et al. (2024) which it shows that integrating user preferences with economic factors results in recommendations that are more in line with user expectations and increases application retention rates. In addition, research by Fahmi (2023) underlines the importance of educating users in understanding the relationship between product features and pricing. Their research shows that providing education-based buying guides can reduce the gap in user expectations. Based on these studies, a more informative and realistic approach, such as the one applied in this recommendation application, has great potential to increase user satisfaction and trust in the recommendation system.

5 Conclusion

This research showed that the smartwatch purchase recommendation application had a fairly good level of precision, especially in adjusting recommendations to the user's budget. The level of user satisfaction with the app was also relatively high, with most users feeling satisfied and willing to recommend the app to others. Results showed that 25.7% of respondents found the application highly relevant to their preferences, 31.4% relevant, and 31.4% somewhat relevant. User satisfaction levels indicated that 34.3% were very satisfied, 34.3% satisfied, and 20% somewhat satisfied, highlighting the application's effectiveness in meeting user expectations. However, there was room to improve the relevance of key feature recommendations and the overall user experience.

To enhance the user experience and satisfaction, it was crucial to improve the algorithm's accuracy to better understand users' specific needs and preferences regarding smartwatch features. Future research should explore hybrid recommendation approaches, such as combining collaborative filtering with KNN or incorporating deep learning-based personalization techniques. Integrating user feedback throughout the app development process would foster trust and ensure the app met user expectations.Additionally, segmenting the market into distinct groups based on demographics, preferences, and needs would enable more personalized and relevant recommendations. Expanding the database with a wider variety of smartwatch models and features would further empower users to find options that best suited their individual preferences, ensuring a comprehensive and tailored experience. Furthermore, this recommendation approach had potential applications in other consumer electronics markets, broadening its impact beyond smartwatches.

Acknowledgements

I would like to express my deepest gratitude to everyone who has contributed to the completion of this work. First and foremost, I thank God Almighty for providing strength, guidance, and blessings throughout this journey. I would like to express my deepest gratitude to my supervisor, Fatchul Arifin, for their invaluable guidance, support, and constructive feedback throughout the completion of this work. I would like to express my heartfelt gratitude to all the participants who kindly took the time to complete the questionnaire for this research. Your valuable input has been instrumental in ensuring the successful completion of this research. Thank you for your contribution and support.

Conflicts of interest

The authors declare that they have no conflict of interest.

References

Abedjan, Z., Chu, X., Deng, D., Fernandez, R. C., Ilyas, I. F., Ouzzani, M., Papotti, P., Stonebraker, M., & Tang, N. (2016). Detecting data errors. Proceedings of the VLDB Endowment, 9(12), 993–1004. https://doi.org/10.14778/2994509.2994518

Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. Big Data, 7(4), 221–248. https://doi.org/10.1089/big.2018.0175

Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. Applied Computing and Informatics, 12(1), 90–108. https://doi.org/10.1016/j.aci.2014.10.001

Chandel, R. S., Sharma, S., Kaur, S., Singh, S., & Kumar, R. (2022). Smart watches: A review of evolution in bio-medical sector. Materials Today: Proceedings, 50, 1053–1066. https://doi.org/10.1016/j.matpr.2021.07.460

Dhanabal, S., & Chandramathi, S. (2011). A review of various k-nearest neighbor query processing techniques. International Journal of Computer Applications, 31(7), 14–22. https://doi.org/10.5120/3836-5332

Ebrahimi, A. (2024). Dynamic User Preferences Optimization in Time-Aware Recommendation Systems. Integration. https://trepo.tuni.fi/handle/10024/160841

Fahmi, S. (2023). Pengaruh Promosi, Harga dan Daya Tarik Produk terhadap Minat Konsumen untuk Beralih menggunakan Sepeda Listrik. JAMIN: Jurnal Aplikasi Manajemen Dan Inovasi Bisnis, 6 (1), 92. https://doi.org/10.47201/jamin.v6i1.199

Ganti, V., & Sarma, A. D. (2013). Data Cleaning. In Synthesis Lectures on Data Management. Springer International Publishing. https://doi.org/10.1007/978-3-031-01897-8

Halder, R. K., Uddin, M. N., Uddin, Md. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. Journal of Big Data, 11(1). https://doi.org/10.1186/s40537-024-00973-y

Ibrahim, M., Bajwa, I. S., Sarwar, N., Hajjej, F., & Sakr, H. A. (2023). An Intelligent Hybrid Neural Collaborative Filtering Approach for True Recommendations. IEEE Access, 11, 64831–64849. https://doi.org/10.1109/access.2023.3289751

Johnson, J. (2021). Designing with the Mind in Mind. Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 1–2. https://doi.org/10.1145/3411763.3444997

King, C. E., & Sarrafzadeh, M. (2017). A Survey of Smartwatches in Remote Health Monitoring. Journal of Healthcare Informatics Research, 2(1–2), 1–24. https://doi.org/10.1007/s41666-017-0012-7

Kishor Bharadwaj, K. S., Jambunath, Y. S., Patil, K. D., Ramnath Babu, T. J., & Santosh Bhargav, D. B. (2024). Preprocessing and Integration of Reproductive Health Data. Data-Driven Reproductive Health, 31–59. https://doi.org/10.1007/978-981-97-7451-7_3

Konstan, J. A., & Riedl, J. (2012). Recommender systems: from algorithms to user experience. User Modeling and User-Adapted Interaction, 22(1–2), 101–123. https://doi.org/10.1007/s11257-011-9112-x

Laaziri, M., Benmoussa, K., Khoulji, S., Larbi, K. M., & El Yamami, A. (2019). Analyzing bootsrap and foundation font-end frameworks: a comparative study. International Journal of Electrical and Computer Engineering (IJECE), 9(1), 713-722. https://doi.org/10.11591/ijece.v9i1.pp713-722

Liu, X., Gao, B., Suleiman, B., You, H., Ma, Z., Liu, Y., & Anaissi, A. (2023). Privacy-Preserving Personalized Fitness Recommender System P3 FitRec: A Multi-level Deep Learning Approach. ACM Transactions on Knowledge Discovery from Data, 17(6), 1–24. https://doi.org/10.1145/3572899

Maillo, J., Ramírez, S., Triguero, I., & Herrera, F. (2017). kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. Knowledge-Based Systems, 117, 3–15. https://doi.org/10.1016/j.knosys.2016.06.012

Mertz, D. (2021). Cleaning data for effective data science: Doing the other 80% of the work with Python, R, and command-line tools. Packt Publishing Ltd.

Murphy-Hill, E., & Murphy, G. C. (2013). Recommendation Delivery. Recommendation Systems in Software Engineering, 223–242. https://doi.org/10.1007/978-3-642-45135-5_9

Nurwanto, F., Ardiyanto, I., & Wibirama, S. (2016). Light sport exercise detection based on smartwatch and smartphone using k-Nearest Neighbor and Dynamic Time Warping algorithm. 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), 1–5. https://doi.org/10.1109/iciteed.2016.7863299

Debora Oktaviani, Fikra Terisha A, Mashita Ayuni, Tesalonika Sembiring, Wynne Lie, & Eryc Yeo. (2024). Analisis Dampak Kecerdasan Buatan dalam Peningkatan Efisiensi Pemasaran Digital di Industri E-commerce Indonesia. JURNAL MANAJEMEN DAN BISNIS EKONOMI, 2(4), 01–10. https://doi.org/10.54066/jmbe-itb.v2i4.2385

Patro, S. G. K., Mishra, B. K., Panda, S. K., Kumar, R., Long, H. V., Taniar, D., & Priyadarshini, I. (2020). A Hybrid Action-Related K-Nearest Neighbour (HAR-KNN) Approach for Recommendation Systems. IEEE Access, 8, 90978–90991. https://doi.org/10.1109/access.2020.2994056

Ramezani, R., Cao, M., Earthperson, A., & Naeim, A. (2023). Developing a Smartwatch-Based Healthcare Application: Notes to Consider. Sensors, 23(15), 6652. https://doi.org/10.3390/s23156652

Ray, R. K., & Singh, A. (2025). From online reviews to smartwatch recommendation: An integrated aspect-based sentiment analysis framework. Journal of Retailing and Consumer Services, 82, 104059. https://doi.org/10.1016/j.jretconser.2024.104059

Reeder, B., & David, A. (2016). Health at hand: A systematic review of smart watch uses for health and wellness. Journal of Biomedical Informatics, 63, 269–276. https://doi.org/10.1016/j.jbi.2016.09.001

Sharma, A., & Amritanshu. (2024). Enhancing Recommendation Systems: A Comparative and Optimization Study of KNN-Based Algorithms. 2024 3rd International Conference for Advancement in Technology (ICONAT), 1–7. https://doi.org/10.1109/iconat61936.2024.10774863

Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. Neurocomputing, 251, 26–34. https://doi.org/10.1016/j.neucom.2017.04.018

Sulistyo, H. A., Kusumasari, T. F., & Alam, E. N. (2020). Implementation of Data Cleansing Null Method for Data Quality Management Dashboard using Pentaho Data Integration. 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 12–16. https://doi.org/10.1109/icoiact50329.2020.9332030

Syriopoulos, P. K., Kotsiantis, S. B., & Vrahatis, M. N. (2022). Survey on KNN Methods in Data Science. Learning and Intelligent Optimization, 379–393. https://doi.org/10.1007/978-3-031-24866-5_28

Triguero, I., García-Gil, D., Maillo, J., Luengo, J., García, S., & Herrera, F. (2018). Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data. WIREs Data Mining and Knowledge Discovery, 9(2). Portico. https://doi.org/10.1002/widm.1289

Usmani, A. A. (2023). Guidelines for Selection of Web Designing Tool & Framework for Web Front-End Application (Doctoral dissertation, Master's Thesis, Tampere University).

Design and implementation of artificial intelligence fusion experimental platform based on machine learning algorithm. (2023). International Journal of New Developments in Engineering and Society, 7(2). https://doi.org/10.25236/ijndes.2023.070202

Wu, L.-H., Wu, L.-C., & Chang, S.-C. (2016). Exploring consumers' intention to accept smartwatch. Computers in Human Behavior, 64, 383–392. https://doi.org/10.1016/j.chb.2016.07.005

Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for kNN Classification. ACM Transactions on Intelligent Systems and Technology, 8(3), 1–19. https://doi.org/10.1145/2990508

Zikry, A., Muhammad Bitrayoga, Siska Yulia Defitri, Akhmad Dahlan, & Nina Dwi Putriani. (2024). Analisis Penggunaan AI dalam Keberhasilan Customer Experience Pengguna Aplikasi E-Commerce Shopee. Indo-Fintech Intellectuals: Journal of Economics and Business, 4(3), 766–781. https://doi.org/10.54373/ifijeb.v4i3.1387