Asia Information System Journal
Volume 2, Issue 2, 46-52
http://ejournal.radenintan.ac.id/index.php/AISJ/index

AISJ Asia Information System Journal
e-ISSN 2963-8593
p-ISSN XXXX-XXXX

TOURIST ATTRACTIONS RECOMMENDER SYSTEM USING COLLABORATIVE FILTERING METHODS AND K-NEAREST NEIGHBORS

Raditya Muhammad*
Universitas Pendidikan Indonesia
Jl. Dr. Setiabudhi No. 229 Bandung, INDONESIA

Fauzan Sukmapratama
Universitas Pendidikan Indonesia
Jl. Dr. Setiabudhi No. 229 Bandung, INDONESIA

Nur Azizah Khoirunnisa
Universitas Pendidikan Indonesia
Jl. Dr. Setiabudhi No. 229 Bandung, INDONESIA

Article Info

Article history:

Received: Aug 31st, 2023
Revised: Nov 7th, 2023
Accepted: Nov 20th, 2023

Keywords:

Collaborative Filtering, K-Nearest Neighbors, Recommender System

Abstract

The many tourist attractions and their various types certainly benefit tourists. Bandung is a famous tourist center and is visited by many domestic and foreign tourists. However, the large number of tourist attractions in Bandung often makes it difficult for tourists to determine their destination, especially if they only have limited time. This research aims to design a tourist attraction recommendation system based on collaborative filtering and K-Nearest Neighbors which can help users by providing recommendations for suitable tourist attractions by displaying seven tourist attraction recommendations. By using collaborative filtering, the system will recommend the best tourist attractions based on ratings and reviews given by users on the internet. Based on the RMSE test results, the train and test data values are in the range of 0.2 - 0.5, which shows that the accuracy of the model is good.

To cite this article:

INTRODUCTION

We have encountered many recommendation systems in everyday life. Applications that provide a recommendation system usually provide a list of recommended items that the user might like or an estimate of how much the user prefers each item. Many applications now have recommendation systems that show users various collections of items, from recommendations for shows, food menus, books, and much more[1][2][3].

The recommendation system can also apply to the selection of tourist attractions. The Greater Bandung area has various potential tourist areas, such as mountains, hot springs, and tea garden tourism, including many food menu choices that arouse tourists' appetites.

The large number of tourist destinations can make tourists confused about their destination[4][5], as happened in research[6]. Sehingga dibutuhkan sistem rekomendasi So a recommendation system is needed that can provide guidance for tourists in choosing tourist locations from various collections of tourist attractions, especially those in the Bandung area.

The potential of tourism as one of the backbones of the economy in Indonesia, of course, means that the tourism sector needs to be managed and processed well, especially in managing digital tourism in the context of the digital economy[7][8].

Previously, Bandung and the majority of tourist areas in Indonesia were in decline due to the impact of the Covid-19 pandemic. In the two years of the pandemic, the number of foreign tourists entering Indonesia has decreased drastically. The number continued to decline until its peak in April 2020, when only 158 thousand foreign tourists came to Indonesia. Not only foreign tourists, local tourist visits also fell by 30%[9].

A tourist attraction recommendation system can be a solution to spark the enthusiasm of tourists to travel by providing recommendations for tourist attractions based

- **Corresponding author:**
Raditya Muhammad, Universitas Pendidikan Indonesia INDONESIA. ✉ radityamuhammad@upi.edu

on the experiences of other tourists who provide reviews on the internet.

The recommendation system works by applying the principles of artificial intelligence (AI) to provide recommendations to users[10]. In general, recommendation systems are classified into three categories, namely content-based filtering (CBF), Collaborative Filtering Systems (CFS), and Hybrid recommendation Systems (HRS)[11][12].

Collaborative filtering is a collaborative filtering technique that utilizes various kinds of information about user behavior, routines, or even choices which will later be accumulated, studied, and made into appropriate recommendations for customers in need based on their similarities with other customers[13]. In this way, the system created can provide recommendations to customers who will choose based on the ratings given by other customers, assuming if someone likes option 1 then he will also like option 2[14].

In classifying tourist objects using the K-Nearest Neighbors (K-NN) concept. K-NN is a method that classifies objects based on the learning data closest to the object. The K-NN algorithm is a classification method for a set of data based on learning from previously classified data. KNN methods and algorithms are included in supervised learning, where the results of new query instances are classified based on the majority of the category closeness distances in K-NN[15][16].

This article discusses recommendations for selecting tourist objects by the system. This selection is not only to find the name of the place and the type of tour but also to consider the ratings and ratings of other visitors who have visited the place. The data source was obtained from open data available on the Kaggle site[17] and this research is a development of research[18].

The system developed helps users who encounter difficulties when checking one the places they want to visit. A recommendation system built based on collaborative filtering and K-Nearest Neighbors with output will display the top seven recommended tourist destinations based on user, rating, and place. We hope that the results of this research can increase the enthusiasm of tourists to travel to various regions in Indonesia, especially in Bandung.

METHOD

The stages carried out are planned activities carried out systematically to achieve the goal. The stages used in this study are shown in Figure 1. The first stage of this research was a literature study to identify problems obtained from journals, scientific articles, and news related to the topic. At this stage, the dataset used in the research is obtained based on data related to the research.

The second stage is data analysis where data preprocessing is carried out, namely removing columns and data that are not used. At this stage, the data is also explored to see the shape and distribution of the data. The third stage is building a recommendation system. At this stage, modeling, training data, and RSME evaluation checks are carried out. The final stage is a recommendation system that has been completed and can display the top seven recommended tourist attractions.

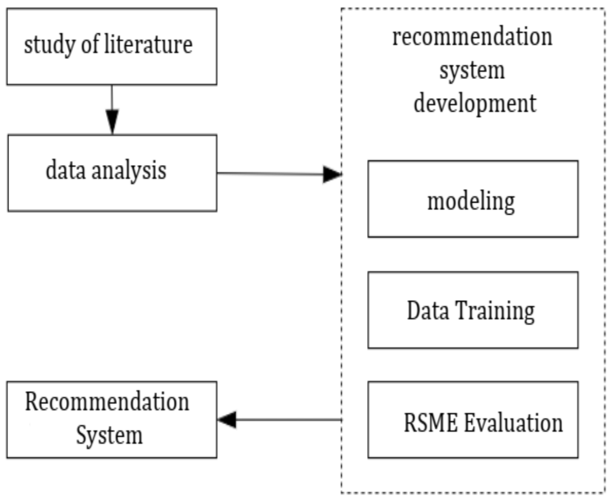


Figure 1. Design System

RESULTS AND DISCUSSION

The set of data structures owned by the tourism destination dataset in Indonesia is divided into three separate datasets, including user data, tourist destination data, and ranking data. Detailed information and structure of each dataset are as follows.

```
place.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 124 entries, 210 to 333
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Place_Id      124 non-null    int64
 1   Place_Name    124 non-null    object
 2   Description   124 non-null    object
 3   Category      124 non-null    object
 4   City          124 non-null    object
 5   Price         124 non-null    int64
 6   Rating        124 non-null    float64
 7   Time_Minutes  50 non-null     float64
 8   Coordinate    124 non-null    object
 9   Lat           124 non-null    float64
 10  Long          124 non-null    float64
dtypes: float64(4), int64(2), object(5)
memory usage: 11.6+ KB
```

Figure 2. Tourist Attractions Data Structure

In Figure 2, the dataset for tourist attractions consists of data, such as place name, category, city, price, rating, and coordinates. Furthermore, the ranking data structure or rating of tourist attractions.

```
[7] rating.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   User_Id         10000 non-null  int64
 1   Place_Id        10000 non-null  int64
 2   Place_Ratings   10000 non-null  int64
dtypes: int64(3)
memory usage: 234.5 KB
```

Figure 3. Rating Data Structure

From Figure 3, it is explained that the rating data structure contains: user ID, place ID, and place rating. Next, the user data structure is explained.

```
user.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   User_Id   300 non-null    int64
 1   Location  300 non-null    object
 2   Age       300 non-null    int64
dtypes: int64(2), object(1)
memory usage: 7.2+ KB
```

Figure 4. Users Data Structure

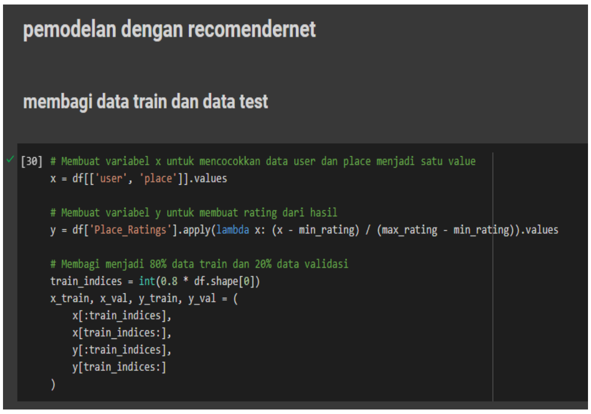
Based on the results of further analysis, it was found that these three data were used as a basis for the proximity of data in the recommendation system modeling process. Below is some information based on the visualization results of each dataset in the data set.

Modeling

Next, system modeling is carried out. The modeling used for the recommendation system in this research uses a deep learning API written in the Python programming language and runs on the Tensorflow machine learning platform.

Some of the steps in the model are as follows:

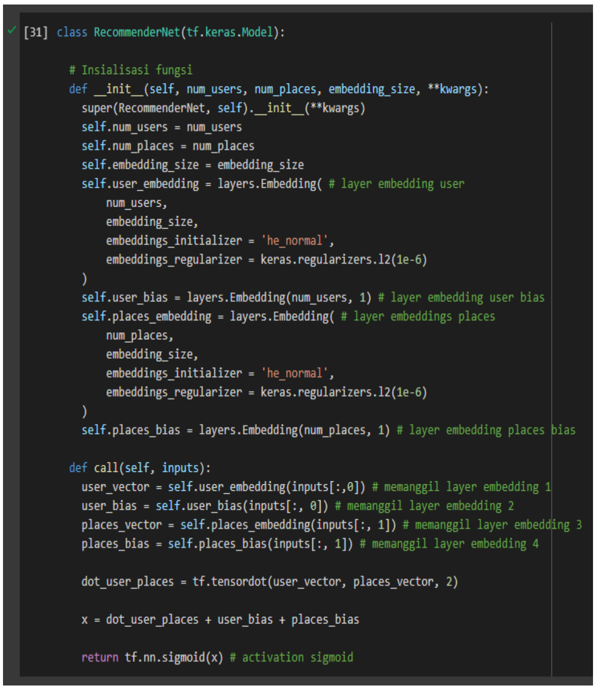
a. Map user ID to "user vector" via matrix embedding
b. Map place_id to "place vector" via matrix embedding
c. Calculate the dot product between the user vector and the place vector, to get a match score between the user and the place (predicted rating)
d. Train to embed through gradient descent using all existing user-place pairs.



```
pemodelan dengan recomendernet

membagi data train dan data test

[30] # Membuat variabel x untuk mencocokkan data user dan place menjadi satu value
     x = df[['user', 'place']].values

     # Membuat variabel y untuk membuat rating dari hasil
     y = df['Place_Ratings'].apply(lambda x: (x - min_rating) / (max_rating - min_rating)).values

     # Membagi menjadi 80% data train dan 20% data validasi
     train_indices = int(0.8 * df.shape[0])
     x_train, x_val, y_train, y_val = (
         x[:train_indices],
         x[train_indices:],
         y[:train_indices],
         y[train_indices:]
     )
```

Figure 5. Prepare data training and validation

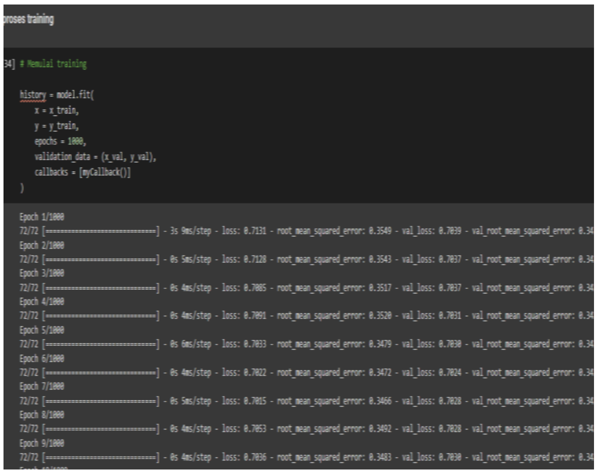
To prepare training and validation data, divide train data and test data. Data is separated into the form of x and y variables, the x variable is used to match the user and place data into one value, while the y variable is used to make a rating of the results. Both data are divided into 80% train data and 20% validation data.



```
[31] class RecommenderNet(tf.keras.Model):

       # Insialisasi fungsi
       def __init__(self, num_users, num_places, embedding_size, **kwargs):
         super(RecommenderNet, self).__init__(**kwargs)
         self.num_users = num_users
         self.num_places = num_places
         self.embedding_size = embedding_size
         self.user_embedding = layers.Embedding( # layer embedding user
             num_users,
             embedding_size,
             embeddings_initializer = 'he_normal',
             embeddings_regularizer = keras.regularizers.l2(1e-6)
         )
         self.user_bias = layers.Embedding(num_users, 1) # layer embedding user bias
         self.places_embedding = layers.Embedding( # layer embeddings places
             num_places,
             embedding_size,
             embeddings_initializer = 'he_normal',
             embeddings_regularizer = keras.regularizers.l2(1e-6)
         )
         self.places_bias = layers.Embedding(num_places, 1) # layer embedding places bias

       def call(self, inputs):
         user_vector = self.user_embedding(inputs[:,0]) # memanggil layer embedding 1
         user_bias = self.user_bias(inputs[:, 0]) # memanggil layer embedding 2
         places_vector = self.places_embedding(inputs[:, 1]) # memanggil layer embedding 3
         places_bias = self.places_bias(inputs[:, 1]) # memanggil layer embedding 4

         dot_user_places = tf.tensordot(user_vector, places_vector, 2)

         x = dot_user_places + user_bias + places_bias

         return tf.nn.sigmoid(x) # activation sigmoid
```

Figure 6. Create a model using recommendernet

After preparing the data, the next step is to create a model using recommendernet. This model calculates the match between the user and the embedding place through the dot product and adds a per-user and per-movie bias. match scores are scaled to [1 : 0] intervals through the sigmoid. At the stage of the training process, model training is carried out based on separated data as shown in Figure 7.



```
proses training

34] # Memulai training

    history = model.fit(
        x = x_train,
        y = y_train,
        epochs = 1000,
        validation_data = (x_val, y_val),
        callbacks = [myCallback()]
    )

    Epoch 1/1000
    72/72 [==============================] - 3s 9ms/step - loss: 0.7131 - root_mean_squared_error: 0.3549 - val_loss: 0.7039 - val_root_mean_squared_error: 0.34
    Epoch 2/1000
    72/72 [==============================] - 0s 5ms/step - loss: 0.7128 - root_mean_squared_error: 0.3543 - val_loss: 0.7037 - val_root_mean_squared_error: 0.34
    Epoch 3/1000
    72/72 [==============================] - 0s 4ms/step - loss: 0.7085 - root_mean_squared_error: 0.3517 - val_loss: 0.7037 - val_root_mean_squared_error: 0.34
    Epoch 4/1000
    72/72 [==============================] - 0s 4ms/step - loss: 0.7091 - root_mean_squared_error: 0.3520 - val_loss: 0.7031 - val_root_mean_squared_error: 0.34
    Epoch 5/1000
    72/72 [==============================] - 0s 6ms/step - loss: 0.7033 - root_mean_squared_error: 0.3479 - val_loss: 0.7030 - val_root_mean_squared_error: 0.34
    Epoch 6/1000
    72/72 [==============================] - 0s 4ms/step - loss: 0.7022 - root_mean_squared_error: 0.3472 - val_loss: 0.7024 - val_root_mean_squared_error: 0.34
    Epoch 7/1000
    72/72 [==============================] - 0s 5ms/step - loss: 0.7015 - root_mean_squared_error: 0.3466 - val_loss: 0.7028 - val_root_mean_squared_error: 0.34
    Epoch 8/1000
    72/72 [==============================] - 0s 4ms/step - loss: 0.7053 - root_mean_squared_error: 0.3492 - val_loss: 0.7028 - val_root_mean_squared_error: 0.34
    Epoch 9/1000
    72/72 [==============================] - 0s 4ms/step - loss: 0.7036 - root_mean_squared_error: 0.3483 - val_loss: 0.7030 - val_root_mean_squared_error: 0.34
```

Figure 7. Training Process

RMSE Testing

Root Mean Square Error or RMSE is a standard way to measure the error of a model in predicting quantitative data. The use of RMSE will show how far the predictions fall from the actual values measured using the Euclidean distance. In this research, RMSE calculations still use the tensorflow library which will be visualized with matplotlib.pyplot.

RMSE is used to determine the magnitude of the deviation formulated in the formula[19].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(RE_i - \overline{RO_I})^2}{n}}$$

After that, the data was tested using 2 comparisons, namely train data and test data, resulting in results as shown in Figure 8.

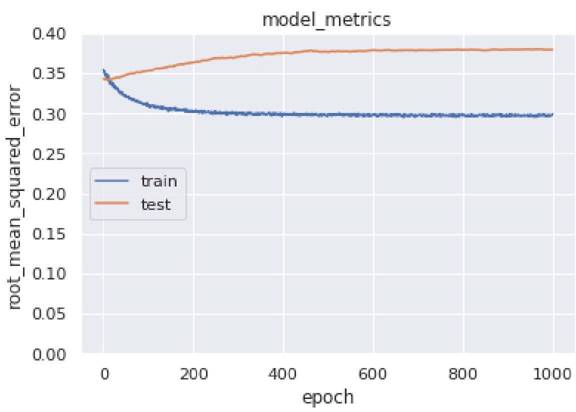


Figure 8. RMSE Testing

Based on the rule of thumb from previous research, a good RMSE value is between 0.2 to 0.5. Therefore, this research model can be said to be good because it is among them.

Furthermore, from the test results, the recommendation system compiles a list of recommendations in Bandung based on user ratings according to the dataset.

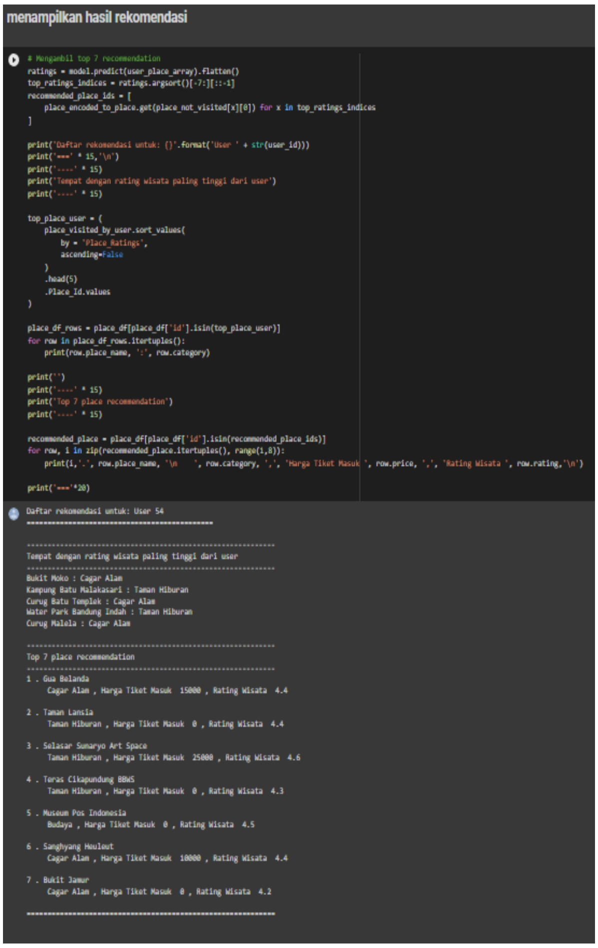


Figure 9. Result of Reccomender System

Based on the output of the recommendation system, the results obtained for the top seven tourist attractions in Greater Bandung are based on user assessments collected from various data on the internet, recommendations for these places can be used as a reference for tourists visiting Bandung. However, please note that the results of this recommendation do not show a comparison of quality between tourist attractions.

This data shows locations that get positive reviews from tourists who have visited the place. The results of this recommendation also do not justify the fact that other tourist attractions that are not included in the recommendation list are bad tourist attractions.

Table 1. Top 7 Places Recommendation

No	Tourist Attraction	Description
1	Selasar Sunaryo Art Space	Park, rating: 4.6
2	Museum Pos Indonesia	Culture, rating: 4.5
3	Gua Belanda	Nature preserve, rating: 4.4
4	Taman Lansia	Park, rating: 4.4
5	Sanghyang Heuleut	Nature preserve, rating: 4.4
6	Teras Cikapundung	Park, rating: 4.3
7	Bukit Jamur	Nature preserve, rating 4.2

Based on the outcome of the recommendation system, the top seven tourist attractions in Greater Bandung were obtained, with a list of Selasar Sunaryo Art Space, Indonesian Postal Museum, Dutch Cave, Elderly Park, Sanghyang Heuleut, Cikapundung Terrace, Mushroom Hill.

The use of a recommendation system provides guidance to users to choose and make decisions based on several tourist options. Recommendation systems imply the benefits that users will obtain according to research[20]. The benefit that users will get by looking at these recommendations is the ease of deciding which tourist attraction to choose without needing to look at reviews and ratings one by one based on the experiences of other users of each tourist attraction, so it is hoped that it can increase the desire to travel, especially in the Greater Bandung area.

The development of tourist attractions recommended by the system has the potential to change along with changes in ratings and reviews of a tourist attraction given by users in the future.

CONCLUSION

Based on the research results, a tourist object recommendation system has been built in the Greater Bandung area which was built by applying AI principles, based on collaborative filtering and K-Nearest Neighbors. Recommendations for tourist attractions are classified based on datasets of tourism destinations in Indonesia which are divided into three datasets, namely: user data, tourist destination data, and rating data. The model in the recommendation system is measured by user, rating, and place.

Based on the results of the RMSE test, the values for the train and test data are in the range of 0.2 - 0.5 to show that the accuracy of the model is good. Furthermore, the system succeeded in providing recommendations for the top seven tourist attractions in the Greater Bandung area.

REFERENCES

[1] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, and F. Ricci, *Recommender Systems Handbook*. 2011. doi: 10.1007/978-0-387-85820-3.

[2] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Appl. Sci.*, vol. 10, no. 21, pp. 1–20, 2020, doi: 10.3390/app10217748.

[3] T. N. Trang Tran, M. Atas, A. Felfernig, and M. Stettinger, "An overview of recommender systems in the healthy food domain," *J. Intell. Inf. Syst.*, vol. 50, no. 3, pp. 501–526, 2018, doi: 10.1007/s10844-017-0469-0.

[4] J. Y. Park and S. C. S. Jang, "Confused by too many choices? Choice overload in tourism," *Tour. Manag.*, vol. 35, pp. 1–12, 2013, doi: 10.1016/j.tourman.2012.05.004.

[5] N. T. Thai and U. Yuksel, "Too many destinations to visit: Tourists' dilemma?," *Ann. Tour. Res.*, vol. 62, pp. 38–53, 2017, doi: 10.1016/j.annals.2016.11.004.

[6] E. Satria, N. Atina, M. E. Simbolon, and A. P. Windarto, "Spk: Algoritma Multi-Attribute Utility Theory (Maut) Pada Destinasi Tujuan Wisata Lokal Di Kota Sidamanik," *Comput. Eng. Sci. Syst. J.*, vol. 3, no. 2, p. 168, 2018, doi: 10.24114/cess.v3i2.9954.

[7] X. Li, X. Liang, T. Yu, S. Ruan, and R. Fan, "Research on the Integration of Cultural Tourism Industry Driven by Digital Economy in the Context of COVID-19—Based on the Data of 31 Chinese Provinces," *Front. Public Heal.*, vol. 10, no. March, pp. 1–10, 2022, doi: 10.3389/fpubh.2022.780476.

[8] I. Heliany, "Wonderful Digital Tourism Indonesia Dan Peran Revolusi Industri Dalam Menghadapi Era Ekonomi Digital 5.0," *Destin. J. Hosp. dan Pariwisata*, vol. 1, no. 1, pp. 21–35, 2019, doi: 10.31334/jd.v1i1.551.

[9] Kemenparekraf, "Strategi Industri Akomodasi Wisata saat Pandemi," kemenparekraf.go.id. Accessed: Aug. 31, 2023. [Online]. Available: https://www.kemenparekraf.go.id/ragam-pariwisata/Strategi-Industri-Akomodasi-Wisata-saat-Pandemi

[10] D. Ramadaniah *et al.*, "Development of Decision Support System Application for Admission of New Students and Determination of Major Using Simple Additive Weighting (Saw)," *Asia Inf. Syst. J.*, vol. 1, no. 2, pp. 70–79, 2022, [Online]. Available: http://ejournal.radenintan.ac.id/index.php/AISJ/article/view/15766

[11] S. Poudel and M. Bikdash, "E Xplaining the P Erformance of C Ollaborative F Iltering M Ethods With," vol. 12, no. 2, pp. 287–295, 2023.

[12] A. R. Fitrianti, A. Rohmani, and W. Widjanarto, "Sistem Rekomendasi Film Berbasis Website Dengan Metode Prototype Menggunakan Algoritma K-Nearest Neighbors (KNN)," *JOINS (Journal Inf. Syst.*, vol. 5, no. 2, pp. 278–287, 2020, doi: 10.33633/joins.v5i2.4168.

[13] I. G. Gusti, M. Nasrun Hasibuan, and R. Astuti Nugrahaeni, "Rekomendasi Sistem Pemilihan Mobil Menggunakan K-Nearest Neighbor (Knn) Collaborative Filtering," *TEKTRIKA - J. Penelit. dan Pengemb. Telekomun. Kendali, Komputer, Elektr. dan Elektron.*, vol. 4, no. 1, p. 26, 2019, doi: 10.25124/tektrika.v4i1.1846.

[14] E. Erlangga and H. Sutrisno, "Jurnal Manajemen Sistem Informasi dan Teknologi Sistem Rekomendasi Beauty Shop Berbasis Collaborative Filtering," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 10, no. 2, pp. 2745–7265, 2020.

[15] E. Riswanto, B. Robi'In, and Suparyanto, "Mobile Recommendation System for Culinary Tourism Destination using KNN (K-nearest neighbor)," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, 2019, doi: 10.1088/1742-6596/1201/1/012039.

[16] A. M. Pravina, “Sentiment Analysis of Delivery Service Opinions on Twitter Documents using K-Nearest Neighbor,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 996–1012, 2022, doi: 10.35957/jatisi.v9i2.1899.
[17] A. Prabowo, “Indonesia Tourism Destination,” kaggle.com. Accessed: Aug. 31, 2023. [Online]. Available: https://www.kaggle.com/datasets/aprabowo/indonesia-tourism-destination
[18] R. I. PUTRA, “Sistem Rekomendasi Destinasi Wisata Kota Bandung,” kaggle.com. Accessed: Nov. 07, 2023. [Online]. Available: https://www.kaggle.com/code/rafkaip/sistem-rekomendasi-destinasi-wisata-kota-bandung
[19] M. A. Azka, P. A. Sugianto, A. K. Silitonga, and I. R. Nugraheni, “Uji Akurasi Produk Estimasi Curah Hujan Satelit Gpm Imerg Di Surabaya, Indonesia,” *J. Sains Teknol. Modif. Cuaca*, vol. 19, no. 2, p. 83, 2018, doi: 10.29122/jstmc.v19i2.3153.
[20] C. H. Yeung, “Do recommender systems benefit users? a modeling approach,” *J. Stat. Mech. Theory Exp.*, vol. 2016, no. 4, p. 43401, 2016, doi: 10.1088/1742-5468/2016/04/043401.