
RECOMMENDATION SYSTEM FOR MOBILE APPLICATION TOUR GUIDE AND TRAVEL SERVICES USING DEMOGRAPHIC FILTERING AND CONTENT-BASED FILTERING METHODS BASED ON ANDROID

Muhammad Tamam Huda¹, Adityo Permana Wibowo²

Universitas Teknologi Yogyakarta, Daerah Istimewa Yogyakarta, Indonesia

tamamhudaaa@gmail.com¹, adityopw@uty.ac.id²

ABSTRACT

This research aims to build a tourism service recommendation system by providing recommendations for tour guides and tourism destinations using machine learning recommendation system methods. The recommendation system method in this research uses the demographic filtering method to recommend tour guides using a model with three stages, namely filtering, scoring and sorting the tour guide data category. At the scoring stage, it is carried out using a decision-making system method using the Simple Additive Weighting method to weight the tour guide value with the weighting value of each criterion in the tour guide data. This research also applies a content-based filtering method using a model with two stages, namely training and recommendation. The training stage uses tf-idf weighting and the recommendation stage uses cosine similarity to recommend tourism spots based on the similarity of metadata for each item. In using the recommendation system, the data will focus on collecting tour guide data and data on tourist attractions selected by the user. This data will be processed by a recommendation system using demographic filtering and content-based filtering methods. Through the Simple Additive Weighting method in demographic filtering and tf-idf weighting with consistent similarity in content-based filtering, this research found personalized recommendation results so that the level of accuracy of the recommendation results becomes more personal and accurate. The results of this research provide a recommendation system for tourist guides and tourist attractions which is implemented into an Android-based mobile application that can be used to meet tourists' needs.

Keywords: tour guide, machine learning, recommendation system, Android, content-based filtering, demographic filtering.

Corresponding Author: Muhammad Tamam Huda

Email: tamamhudaaa@gmail.com



INTRODUCTION

The tourism industry in Indonesia experiences progress every year. Based on September 2023 Tourism Developments, the Central Statistics Agency (BPS) stated that domestic tourism trips in the third quarter of 2023 were recorded at 192.52 million trips. This number increased by 13.36 percent compared to the third quarter of 2022. This figure is also higher by 15.07 million trips or a growth of 8.49 percent compared to the number of trips in the third quarter before the COVID-19 pandemic in Indonesia. Apart from that, foreign tourist arrivals in Indonesia reached 1.07 million visits and experienced an increase of 52.76 percent compared to last year. Based on these data observations, tourism travel in Indonesia is increasing every year, this has caused an increase in tourists' needs for tourism trips.

The needs of tourists when going on tourism trips include using tourism travel services, especially for foreign tourists who need information on tourist attractions such as location, prices and comprehensive information about tourist attractions. The large amount of information that

must be sought makes foreign tourists and even domestic tourists confused in determining tourist attractions as tourism travel destinations. This is also inseparable from the need for tour guides when carrying out tourism trips. Tour guides are very important for tourists when going on tourism trips, especially for tourist attractions that have just opened. The tour guide will be a travel partner and help tourists identify tourist attractions. With this research, the author will build an application using a recommendation system that can help tourists find information on tourist attractions and can plan tourism trips with tourist destinations that can be determined by the user. Apart from that, the system being built will provide tourist guide recommendations to users.

Machine Learning-based recommendation systems are advanced engines that use machine learning algorithms to segment customers based on their user data and behavioral patterns (such as browsing history, likes, or reviews) and target them with personalized information or content suggestions. Recommendation systems rely on machine learning to process huge customer data sets and consider a wider range of parameters to perform classification and targeting processes. Such models are usually used for predictive analysis in making recommendations. Thus, this research will create an application that applies a recommendation system algorithm using the demographic filtering method with a decision support system using the simple additive weighting method which will carry out predictive analysis of data sets from users to determine Tour Guide recommendations. The recommendation system also uses a content-based filtering method with tf-idf vectorizer bottling and cosine similarity which is used to provide recommendations for tourist attractions based on various categories. In implementing the recommendation system in this tourism service application, it can provide recommendations to tourists in meeting their needs for tourism trips.

Table 1. Research Comparison Table

Title	Author (Year)	Platform	Recommendation Method
Development of a Tour Travel Service Information System and Tour Package Recommendation System in Yogyakarta Using the Content-Based Filtering Method	Anggun Dwi Cahyadi (2020) (Cahyadi, 2020)	Web	Content-Based Filtering
Design and Development of an Android Mobile Application for the Gresik Regency Tourist Location Information System Using the Item-Based Collaborative Filtering Method	Hendry Hermawan (2019) (Hermawan & Rosyid, 2021)	Mobile Android	Collaborative Filtering
Content-Based and Collaborative Filtering on Recommended Tourism Destinations in the Yogyakarta Region	Aprilia Saptu Ningrum, Heru Cahya Rustamaji and Yuli Fauziah (2019) (Ningrum et al., 2019)	Mobile Android	Hybrid Filtering (Content-Based and Collaborative Filtering)
Application Development Travelbuteng.Id For Promotion Central Buton Regency Tourism (Module Based Mobile Application Android)	Muhammad Ammar, Umar Ali Ahmad, Burhanuddin Dirgantoro, Randy Erfa Saputra, Reza Rendian Septian (2022) (Ammar et al., 2022)	Mobile Android	Knowledge-Based, Collaborative Filtering, Content-Based Filtering, and Hybrid Filtering

Application of Content-Based Filtering Algorithms for Destination Recommendations	Aprianto (2022) (Aprianto, 2022)	Mobile Android	Content-based Filtering
---	----------------------------------	----------------	-------------------------

Travel on the Picnicker Application

Based on Table 1, Based on Table 1, each previous study has a different recommendation system method and uses data to produce recommendations. (Cahyadi, 2020), conducted research on tourism package recommendations for the system to be built with a problem formulation of how to build a travel package recommendation system in Yogyakarta for website service users using the Content-Based Filtering method with the Nearest Neighbor algorithm approach. Where the input data is data on tourist attractions to consumers based on categories and assessments based on attractions with tourist objects for each category. This data will be processed by the system so that it can provide output in the form of tour package recommendation data to users.

(Hermawan & Rosyid, 2021), conducted research on tourist location information systems. This research has a problem formulation on how to apply the item-based Collaborative Filtering method to the Android tourism application information system in Gresik Regency. Where the input data from the system that will be used is based on the location and rating of tourist attractions. This data will be processed for assessment using ratings, making predictions through ratings, and calculating similarity values with a weighted sum to produce output recommendations for tourist attractions using Collaborative Filtering.

(Ningrum et al., 2019), conducted research into the development of a tourism destination recommendation system. This research has a problem formulation of how to build a system that is able to provide recommendations for tourist attractions. This recommendation system uses the Hybrid Filtering method, combining two methods, namely the Collaborative Filtering and Content-Based Filtering methods. The input data from the system that will be used is based on tourist facilities, entrance ticket prices and tourist categories. This data will be processed by the system to get a rating using the Content-Based method and the rating prediction calculation process is carried out using Collaborative Filtering, so that it will provide output in the form of tourist recommendations using Hybrid Filtering.

(Ammar et al., 2022), conducted research regarding the development of the Travelbuteng.Id application for tourism promotion in Central Buton Regency. This research aims to improve the tourism sector which can also help the economy there. The recommendation system in this research uses several methods, namely Knowledge Based, Collaborative Filtering, Content Based Filtering and Hybrid Filtering. The data input used utilizes feedback from users, price, type of tour, and day. This data will be processed by the system to get recommendations in the form of tour packages based on the level of similarity.

(Aprianto, 2022), conducted research on the application of content-based filtering algorithms for recommending tourist destinations in the Picknicker application. This research was conducted to provide a platform that can accommodate tourists' needs in reaching information on local tourist destinations directly. The system used in this research uses a recommendation system with a content-based filtering method. The input data used is between other tourist destinations, tourist locations and tourist descriptions to determine the highest level of similarity using the content-based filtering method so that it can produce recommendations for tourist attractions.

There are differences between previous research and the research that the author will carry out. Previous research focused on meeting tourists' needs by providing information on tourist attractions through a recommendation system. This research utilizes information about tourist attractions using a recommendation system to provide recommendations for tour guides. In this research, the system is also used to meet tourists' needs in planning tourist trips by determining tourist destinations according to tourists' wishes. The difference between previous research and the research to be conducted is in the method of building a recommendation system. In this research, the system built uses the demographic filtering method with alternative value weighting using simple additive weighting and content-based filtering with document weighting using tf-idf vectorizer and similarity level. documents with cosine similarity. Differences also exist in the input data used, in research the data used include tourist categories, tourist metadata, tour guide categories and tour guide values. Apart from that, the resulting output is also different. In this research, the recommendation system produces recommendations for tourist attractions and tourist guides which are applied to the tourism trip planning system.

This research aims to create an application, "Tour Guide and Tourist Travel Services," to overcome the problems above. There are two objectives of this research, namely: 1) To produce an application that can be used to recommend tourist attractions and tour guides using a machine learning recommendation system, and 2) To produce the accuracy of a machine learning recommendation system in recommending tourist attractions and tour guides. It is hoped that this research will be useful for users of the "Tour Guide and Tourist Travel Services" application. It is hoped that this research can create a tourism agenda according to the desired or recommended tourist attractions get recommendations for competent tour guides, and can organize visits to tourist destinations with a competent tour guide.

METHOD

According to (Falk, 2019), a recommendation system is a concept for calculating and providing relevant content to users based on user knowledge, content, and interactions between users and items. Meanwhile, according to (Aggarwal, 2016) defines a recommendation system as a system that utilizes various data sources to conclude customer interests. The entity to which a recommendation is given is called a user, and the product being recommended is called an item. The basic principle of recommendation is that there is a significant dependency between the user and the item-centric activities, so the recommendation analysis is based on previous interactions between the user and the item. Recommendation systems work using two types of data, namely user interactions with items and information about users and items (Yoshua & Bunyamin, 2021) (Huda et al., 2022). Systems that use the first type of data are called Collaborative Filtering methods, while systems that use the second type of data are called Content-Based Filtering methods. Systems that combine the two methods are called Hybrid Recommendation Systems (Wijaya & Alfian, 2018) (Prasetyo et al., 2019).

According to (Müller & Guido, 2016), Machine Learning is machine learning by extracting knowledge from data in research fields, including the intersection of statistics, artificial intelligence, and computer science. It is also known as predictive analytics or statistical learning. Meanwhile,

according to (Pajankar Joshi, 2022), Machine Learning is the science of making computers capable of acting based on available data without being explicitly programmed to act in a certain way. (Raschka & Mirjalili, 2019) (Pratama et al., 2023) define Machine Learning as the application and science of algorithms that understand data using self-learning algorithms so that they can transform data into knowledge, utilize algorithms to find patterns in data, and make predictions about future events.

According to (Aggarwal, 2016), Demographic Filtering is demographic information about users that is obtained to learn classifiers that can map certain demographics to ratings or buying tendencies. Meanwhile, according to (Syahreza & Tanjung, 2018) (Nanda, 2023), Demographic Filtering is a system that uses demographic data such as age, gender, education, etc., to classify user groups. According to (Negre, 2015), Content-Based Filtering is a system that recommends items according to the user's profile based on items that the user has liked in the past or interests that are explicitly determined by the user (Aisha, 2022) (Hutabarat, 2022). Meanwhile, according to (Gorakala & Usulli, 2015), Content-Based Filtering is a system that recommends items to users by considering the similarity of items and user profiles; the system recommends items that are similar to those that users have liked in the past.

According to (Aprilian Saputra, 2020), Simple Additive Weighting is a weighted addition method. The basic concept of the SAW method is the weighted sum of the performance ratings for each alternative on all criteria. According to (Hyde, 2020), UML determines use cases to describe system functionality in accordance with requirements. Use case diagrams to determine what the system does from the perspective of an external observer; only determine what the system does, not how it does it. A use case diagram contains three elements: actors, communication links (or associations), and the actual use case. According to (Yen, 2019) Entity Relationship Diagram (ERD) is a tool for planning and designing databases, especially used in conjunction with data normalization. Entity relationship models start with entities, data normalization starts with attributes, and both tools tend to verify each other. Entity model entities, attributes, and relationships map seamlessly to physical databases. Entity Relationship Diagram has a concept for planning and designing databases and for modeling system data (Ramadhan & Purwandari, 2018) (Pulungan et al., 2023). According to (Thanaki, 2017), the TF-IDF concept is the inverse frequency document frequency term, which is a numerical statistical field for deciding how important a word is to a particular document in the current data set or corpus.

The functionality of the system depends on the data it uses. In this research, dummy data is used to implement a recommendation system, including tour guides and tourism data obtained from tourism media. This data set includes 1) Tour Guide Data, containing guide details such as name, email, contact number, and status; and 2) Tourism data sourced from tourism media, in the form of information on site names, categories, facilities, descriptions, and operating hours. Data collection is divided into primary data, which is obtained directly from tourists, and secondary data, which is obtained indirectly through various media sources to complement and strengthen primary data, thereby ensuring higher research validity. Data collection methods include interviews, structured to understand respondents in depth through face-to-face or telephone interactions; administration of questionnaires directly related to the needs of tour guides; and literature reviews, which draw

theoretical concepts from scientific works, newspapers, and magazines. Collection was carried out in two stages, namely November to December 2022 for primary and secondary data, followed by February to May 2023 to complete secondary data. This study aims to update tourism services such as Traveloka, which currently needs more flexibility for multi-destination travel and faces challenges in providing tour guide services. Moreover, intends to build a new computerized system to increase tourist satisfaction.

In planning the application system to be built, it is based on previous research. In the research, the system that will be built is used to meet tourists' needs in planning tourist trips by determining tourist destinations according to the user's wishes. The system that will be built will also provide recommendations for tourist attractions and tourist guide recommendations. In building a recommendation system, this research uses demographic filtering and content-based filtering methods. The application of the demographic filtering method uses a model with three stages. The first stage is filtering, the data will be filtered to obtain data according to user preferences. The second stage is scoring, the data will be weighted by alternative values using simple additive weighting to get the preference value for each alternative. The third stage is sorting, the results of the scoring stage will sort the data according to the highest value to be recommended to users. The application of the content-based filtering method uses a model with two stages. The first stage is training, the data will be weighted by documents using a tf-idf vectorizer to calculate how important each word that appears in each document is. The second stage is recommending, at this stage it is carried out to produce recommendations by comparing the level of similarity between documents using the cosine similarity method.

RESULTS AND DISCUSSION

The back-end system for the Tour Guide and Travel Service application was built using the Django framework based on the Python programming language. The following is the implementation of the recommendation system in the Tour Guide Service application using the Python programming language.

Demographic Filtering

The demographic filtering recommendation system has three stages, namely filtering, scoring, and sorting. Implementation of a recommendation system using the demographic filtering method is applied to determine tour guide recommendations based on data obtained from users. The following is the implementation of the demographic filtering model:

Filtering

At the filtering stage, the first stage is determining the requirements or conditions before filtering is carried out. Demographic filtering in determining tour guide recommendations uses conditions with input of tour category and tour guide status in state 1 (ready). If the tour guide status is 0, then it is not included in the filtering requirements. The following is the application of the filtering stages in the demographic filtering model.

```
def filtering(self):
    try:
        if self.__serializerField.is_valid():
            df = pd.DataFrame(list(self.__categories.all().values()))
```

```

df_filter_ktg = df[df[self.__travel_category].all(axis=True)]
category_filter_id = df_filter_ktg['tg_id'].tolist()
filter_stts = self.__tourguide.filter(
id_tg__in=id_filter_category, status=False)
self.__id_filtering = pd.DataFrame(list(filter_stts.values()))[
'id_tg'].tolist()
return self.__id_filtering
else:
return self.errors
Except Exception as e:
print(e)

```

The application of the filtering stages above is carried out using the Pandas library with the condition that the tour guide data will be filtered based on tourist categories with tour guide status 1 (true). After filtering, the tour guide data will be returned based on the tour guide ID according to the filtered data. This data will then be used at the scoring stage.

Scoring

The scoring stage determines the suitability of the tour guide to be recommended to users. To perform scoring on demographic filtering, researchers used the Simple Additive Weighting (SAW) method. This method has several stages, namely determining the weight of each criterion, calculating the normalization matrix, and calculating the final value of each alternative. The following is the implementation of the scoring stages in the demographic filtering model.

1) Determining the Weight of Each Criteria

```

def scoring(self):
try:
if self.__serializerField.is_valid():
values_tg = self.__values.filter(
tg_id__in=self.__id_filtering).values()
df_values = pd.DataFrame(list(values_tg)).set_index('tg_id')
data = df_values.transpose().to_dict()
C_MaxMin = []
for key in self.__criteria.keys():
C = [data[i][key] for i in data.keys()]
if self.__criteria[key]['attribute'] == 'benefit':
C_MaxMin.append(max(C))
Else:
C_MaxMin.append(min(C))
weight = [self.__criteria[i]['weight']]
for i in self.__criteria.keys():
norms = []
v = []
For values in data.values():
norms = []
i = 0
vn = 0
For key, val in value.items():
if self.__criteria[key]['attribute'] == 'benefit':
n = val/C_MaxMin[i]
Else:
n = C_MaxMin[i]/val
vn += (n * weight[i])
norm.append(n)
i += 1
vn_format = "{:.2f}".format(vn)
norms.append(norm)
v.append(vn_format)
matrix_norms = np.array(norms)
self.__scoring = {}
i = 0
for key in data.keys():
self.__scoring[key] = v[i]
i += 1
return self.__scoring

```

```

else:
    Return self. Errors
except Exception as e:
    print(e)

```

The implementation of the Python programming language above is an implementation for determining the weight of each criterion. If the criteria attribute is a benefit, then the max value of the criteria weight will be used for normalization. If the attribute is a cost, then the min value of the weight value will be used for normalization. These provisions are obtained from the equations specified in the simple additive weighting method.

2) Calculate the normalization matrix and preference value for each alternative

```

norms = []
v = []
for values in data.values():
    norms = []
    i = 0
    vn = 0
    For key, val in value.items():
        if self.__criteria[key]['attribute'] == 'benefit'
            n = val/C_MaxMin[i]
        Else:
            n = C_MaxMin[i]/val
        vn += (n * weight[i])
    norm.append(n)
    i += 1
    vn_format = "{:.2f}".format(vn)
    norms.append(norm)
    v.append(vn_format)
matrix_norms = np.array(norms)
self.__scoring = {}
i = 0
for key in data.keys():
    self.__scoring[key] = v[i]
    i += 1
return self.__scoring
else:
    Return self. errors
except for Exception as e: print(e)

```

The implementation of normalization calculations is carried out by calculating the value of the criteria and using the equation. If it is of the cost type, then the value is divided by the max value, and if it is the cost type, then normalization is carried out using the equation of the min value divided by the value. After getting the normalization matrix, the preference value is calculated for each. The results after calculating alternative preferences, the data will be returned in the form of an object that has a tour guide ID value and a preference value. Then, the data will be sorted or sorted.

Sorting

The final stage in finding recommendations using demographic filtering is sorting to get the first order or those with the greatest value. The sorting implementation is in the source code below.

```

def sorting(self):
    try:
        if self.__serializerField.is_valid():
            data_sorted = sorted(
                [(key, value) for (key, value) in self.__scoring.items()], reverse=True)
            self.__sorting = pd.DataFrame(
                data_sorted, columns=['tg_id', 'rating']).to_dict(orient='records')
            return self.__sorting
        else:
            Return self. errors
    Except Exception as e:
        print(e)

```


Implementation of sorting is carried out after obtaining the value results from calculations using SAW. This sorting will determine the final recommendation results using demographic filtering. The sorting results from the largest value are the recommendation results.

Content-Based Filtering

The content-based filtering recommendation system also uses a model with two stages, namely training and recommendation. The implementation of content-based filtering is used to make tourist recommendations based on the similarity of data obtained from users. The following is the implementation of the content-based filtering model:

a. Training

The first stage in content-based filtering recommendations is carried out with training data. At this stage, the process carried out creates metadata as the content of an item that will look for similarities in the content. After getting the metadata, text processing is then carried out. Next, the metadata will be used to perform the TF-IDF Vectorizer as an encoder.

```
def fit(self):
    # Create metadata with data on facilities, attractions, addresses, and opening hours.
    self.data['metadata'] = self.data.apply(lambda row: f'{',
    '.join(row['facilities']), {row['interest']], {'', '. join(row['category'])}},
    {row['address']], {row['opening_hours']}", axis=1)
    self.data['metadata'] = self.data['metadata'].str.replace(", ", " ")
    # TF-IDF Vectorizer
    corpus = self.data['metadata']
    encoder = TfidfVectorizer(stop_words='english')
    self.bank = encoder.fit_transform(corpus)
```

The data training stage of the content-based filtering model creates metadata, which is a combination of several data from an item, including data on facilities, categories, addresses, and opening hours. The metadata obtained will be subjected to text processing to remove symbols or punctuation, and then the metadata will be used as a corpus. The next stage is to encode using TFIDF Vectorizer using the Skelearn library. The results of the TFIDF Vectorizer will be stored in bank variables.

b. Recommend

The recommendation stage is used to carry out the recommendation process using cosine similarity with linear carnal on the TFIDF Vectorizer bank. After obtaining the cosine similarity matrix, the next step is to find the pairwise similarity score from the cosine similarity matrix. From the results obtained in the pairwise similarity score, a sorting process is then carried out to sort the similarity scores from closest to furthest. The implementation of the recommendation stage can be seen in the following source code.

```
def recommend(self):
    # Get the index that matches the ID content
    idx = self.indices[self.content]
    # Compute cosine similarity matrix using a linear kernel
    cosine_sim = linear_kernel(self.bank, self.bank)
    # Get the pairwise similarity scores
    sim_scores = list(enumerate(cosine_sim[idx]))
    # Sort based on the similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    # Get the scores for the ten most similar movies
    sim_scores = sim_scores[1:11]
    # Get indices
    _id = [i[0] for i in sim_scores]
    # Return the most similar recommendations
    self.result = self.data_user.loc[_id]
    self.result['score'] = [i[1] for i in sim_scores]
```

```
def result(self):  
    self.fit()  
    self.recommend()  
    return pd.DataFrame.to_dict(self.result, orient='records')
```

The initial stage of recommendation is carried out by getting the index of an item. Next, calculate cosine similarity using linear channels for banks. The index obtained is used to obtain a pairwise similarity score, where the similarity calculation of an item will refer to the index of the item or document. Then, the results of calculating document similarity based on the similarity score will be sorted to get 10 item recommendations that will be given to users.

Results of Implementation of Demographic Filtering

The Tour Guide and Travel Service Application uses the demographic filtering method to determine tour guide recommendations for users. Demographic filtering in the system requires input items in the form of tourism categories. The input data obtained will then be used to determine tour guide recommendations. The results of demographic filtering recommendations for tour guides can be seen in the implementation of the following application.

1) Add Excursions to Plan a Trip

Users add items in the form of information about tourist attractions. These items will be used to make trip plans.

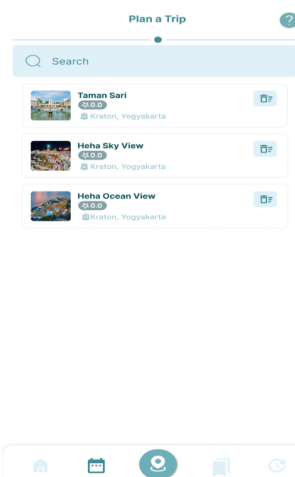


Figure 1. Plan a Trip page

In the results of this implementation, the user adds the items that have been added, namely Taman Sari and Heha Sky View Heha Ocean View, to the trip planning, which can be seen in Figure 1.

2) Choosing a Tour

The stage of choosing a tour is that the user determines which tours will be included in the trip planning. The selected items will be used to determine tour guide recommendations based on the category of the item or tour. After determining the item, the user presses the "Plan Now" button and then moves to the trip details confirmation page.

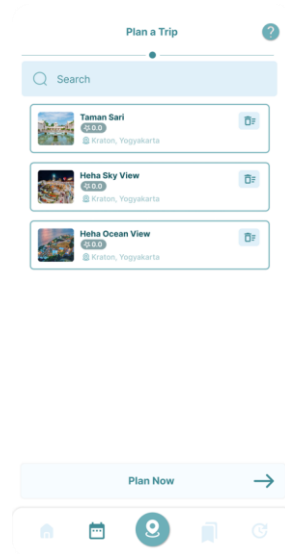


Figure 2. Choosing a Tour

3) Confirm Trip Details

Trip detail display where the user must set data and confirm data from trip planning. The data starts from tourist destinations, dates, passengers, and prices. After ensuring the data is correct, the user presses the "Search Tour Guide" button to find recommended tour guides.

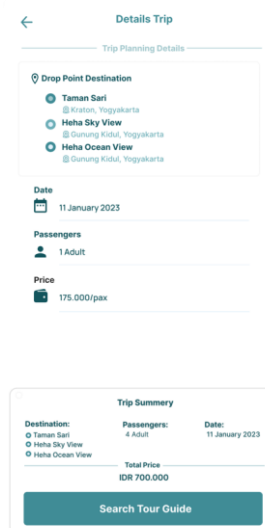


Figure 3. Confirm Trip Details

4) Get Tour Guide Recommendations

The results of trip planning were that at this stage, the user had received a recommended tour guide using demographic filtering. At this stage, the application displays details of the trip that has been made, starting from the destination, trip status, passengers, date, and information on the results of the tour guide's recommendations. In the picture above, the user gets a recommended tour guide with a rating of 4.7. The results of trip planning also produce a QR Code, which can be scanned in the application and will display detailed trip information that the user has made.

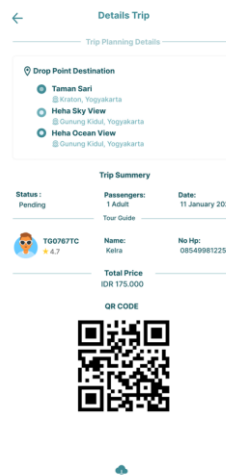


Figure 4. Getting Tour Guide Recommendations

Results of Implementation of Content-Based Filtering

The implementation of content-based filtering is used to recommend tourist attractions for users. Recommendations are obtained by accessing the details of a tourist attraction; then, the system will provide recommendations in the form of tourist attractions that have similar content or metadata according to the tourist attractions accessed by the user. The results of applying content-based filtering to one of the tourist attractions accessed by users, namely Ngrenahan Beach. Next, the system will carry out a recommendation process with items or tourist attractions that have the same level of similarity as Ngrenahan Beach. Recommendations for content-based filtering that have a level of similarity close to Ngrenahan Beach are Ngetun Beach, Sadranan Beach, and Krakal Beach.

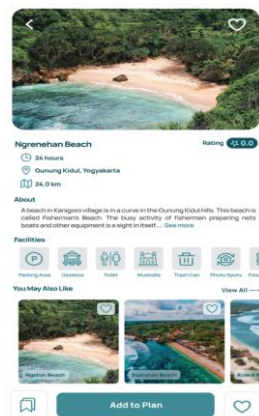


Figure 5. Content-Based Filtering Tourism Recommendations

Implementation of Demographic filtering

Implementation of demographic filtering and simple additive weighting using the Python programming language. The demographic filtering recommendation system has three stages, namely filtering, scoring, and sorting. The following is a discussion of the results of implementing demographic filtering in the Tour Guide Service application system.

a. Retrieving Data from Database

When implementing a recommendation system using demographic filtering, the required data is taken from the database server. This data will be used as information that will be analyzed

in implementing demographic filtering. The data that will be used in the demographic filtering model uses tour guide information data, including tour guide categories and tour guide values. The tour guide category data will be used to carry out the filtering process; then, the tour guide value data will be used to carry out the scoring and sorting process.

Table 2. Tour Guide Category Data

Id_tg	K1	K2	K3	K4	K5	K6	K7	S
TG0251GG	1	1	1	0	0	1	1	1
TG0261SG	1	1	1	1	1	1	0	1
TG0564SG	0	1	0	1	1	0	1	1
TG0711TC	1	1	1	1	0	1	0	1
TG0767TC	1	1	1	1	0	1	0	1
TG0897GG	1	0	1	1	0	1	0	1

Table 2 Description:

K1 = nautical

K2 = culture

K3 = agrotourism

K4 = nature reserve

K5 = history

K6 = nature

K7 = religion

S = state

Data calls are carried out in the Python programming language using the Django query set. The tour guide data will be used at the filtering stage.

b. Filtering

The first stage in filtering is determining the requirements or conditions before filtering is carried out. Demographic filtering in determining tour guide recommendations uses conditions with input of tour category and tour guide status in state 1 (ready). If the tour guide status is 0, then it is not included in the filtering requirements. The application of conditions or requirements can be seen in Table 3.

Table 3. Filtering Terms

Variable	Mark
Status	1
Category	Nature, Marine

Implementation of filtering with tourism categories, namely natural and marine, then tour guide status 1. Filtering is carried out using the Pandas data frame library. The implementation results above produce data, as shown in Table 4.

Table 4. Tour Guide Filtering Results Data

Id_tg	nautical	natural
TG0251GG	1	1
TG0261SG	1	1
TG0711TC	1	1
TG0767TC	1	1
TG0897GG	1	1

The results of the filtering stage where the data displayed are tour guide category data which has a value of nautical = 1, nature = 1, and status = 1; it can be seen that this tour guide

handles tourism in the nautical and nature categories with status 1 (ready). This data will be used as an alternative at the scoring stage.

c. Scoring

The scoring stage determines the suitability of the tour guide to be recommended to users. To perform scoring on demographic filtering, researchers used a decision support system method, namely Simple Additive Weighting (SAW). This method has several stages, namely determining the weight of each criterion, calculating the normalization matrix, and calculating the preference value for each alternative.

a) Determining the Weight of Each Criteria

The weight of each predetermined criterion is used to calculate SAW. There are four criteria for assessing tour guides, namely personality, skills, and knowledge. These four criteria have a benefit criteria attribute and a criteria weight of 1.25 each. In determining the weight of each criterion using Python, if the criterion attribute is a benefit, then the max value of the criterion weight will be used for normalization. If the attribute is a cost, then the min value of the weight value will be used for normalization. The weight of each criterion that will be applied can be seen in Table 5.

Table 5. Weight of Each Criteria

Criteria	Criterion Name	Criteria Attributes	Criteria Weight
C1	Personality	Benefits	1.25
C2	Skills	Benefits	1.25
C3	Knowledge	Benefits	1.25
C4	Responsibility	Benefits	1.25

b) Tour Guide Value Analysis Stage

At the Analysis stage, namely changing the criteria weight values in the tour guide data as an alternative. Retrieve the tour guide value in the tour guide value table in the database based on data that has been filtered. This alternative value will be used for matrix normalization. Tour guide value data can be seen in Table 6.

Table 6. Weight Value for Each Alternative Criteria

alternative	c1	c2	c3	c4
TG0251GG	85	85	90	80
TG0261SG	90	80	80	85
TG0711TC	80	90	90	80
TG0767TC	85	85	85	90
TG0897GG	90	80	85	85

c) Calculating Matrix Normalization

Python implementation in calculating normalization and preference values using the SAW method. Normalization is carried out by calculating the value of the criteria and using the equation. If it is of the cost type, the value is divided by the max value, and if it is the cost type, then normalization is carried out using the equation of the min value divided by the value. The following normalization calculation results are in Table 7.

Table 7. Normalization Matrix Results

0.9444	0.9444	1	0.8889	0.9444
1	0.8889	0.8889	0.9444	1

0.8889	1	1	0.8889	0.8889
0.9444	0.4444	0.9444	1	0.9444
1	0.8889	0.9444	0.9444	1

d) Calculating Alternative Preference Values

The final stage is a ranking by calculating the preference value for each alternative, which is obtained by multiplying the weight value of each criterion with the normalized performance rating value. The results of calculating alternative preference values can be seen in Table 8.

Table 8. Calculation Results of Alternative Preference Values

alternative	mark
TG0251GG	4.72
TG0261SG	4.65
TG0711TC	4.72
TG0767TC	4.79
TG0897GG	4.72

d. Sorting

The final stage in finding recommendations using demographic filtering is sorting to get the first order or those with the greatest value. Sorting is done after getting the value results from calculations using SAW. This sorting will determine the final recommendation results using demographic filtering. The results of the sequencing that has been carried out can be seen in Table 9.

Table 9. Recommendation Sorting Results

alternative	mark
TG0767TC	4.79
TG0897GG	4.72
TG0711TC	4.72
TG0251GG	4.72
TG0261SG	4.65

It can be seen that the alternative with the highest value is the tour guide who has the ID "TG0767TC". These results will be used to return tour guide data as a result of recommendations obtained by the user.

Application of Content-Based Filtering

The application of content-based filtering in recommending tours uses a model with two stages, namely training, and recommendation. At the training stage, TFIDF Vectorize is carried out; then, at the recommendation stage, the cosine similarity calculation is carried out. The following is a discussion of the results of implementing content-based filtering to provide tourist recommendations on the Tour Guide and Travel Service application.

1) Retrieving Data from Database

Implementing a content-based filtering recommendation system requires data taken from a database server. This data will be used as information to carry out the recommendation process using a content-based filtering model. The data used in the content-based filtering model uses tourist information data, which will be used as metadata, including facilities, attractions, addresses, and opening hours. In this discussion, the application of content-based filtering is

carried out to get recommendations from Ngrenahan Beach tourism with tourist data, including Ngetun Beach, Sadranan Beach, and Krakal Beach. From the four tourist data, a recommendation process will be carried out using a content-based filtering model. The tourism data that will be used can be seen in Table 10.

Table 10. Tourism Data

Name	Facility	Attractiveness	Category	Address	Opening hours
Ngetun Beach	parking area, prayer room, photo spots, trash cans, toilets, food stalls	swimming, sunrise and sunset views, camping, fishing, turtle viewing	maritime	Gunung Kidul, Yogyakarta	24 hours
Sadranan Beach	parking area, gazebo, homestay, prayer room, snorkeling, photo spots, trash cans, toilets, food stalls	snorkeling, sunbathing, white sand, camping, nautical	maritime	Gunung Kidul, Yogyakarta	24 hours
Krakal Beach	parking area, homestay, prayer room, snorkeling, photo spots, surfing, trash cans, toilets, food stalls	swimming, snorkeling, surfing, camping, culinary, restaurants, white sand, big waves	maritime	Gunung Kidul, Yogyakarta	24 hours
Ngrenahan Beach	parking area, gazebo, prayer room, photo spots, trash cans, toilets, stalls	coral hills, cultural tourism, swimming, camping,	maritime	Gunung Kidul, Yogyakarta	24 hours

2) Training

At the training stage, the tourism data that has been obtained will be used to create metadata for each item; then, the metadata will be used as a corpus to carry out TFIDF Vectorize.

a. Creating Metadata

Metadata is the content that will be used as a corpus to carry out TFIDF Vectorize. Metadata is formed by combining data on facilities, attractions, categories, addresses, and opening hours. The results of combining the data into metadata that will be used as a corpus can be seen in Table 11.

Table 11. Corpus Content-Based Filtering Data

Name	Corpus
Ngetun Beach	parking area prayer room photo spots trash cans toilets food stalls swimming sunrise and sunset views camping fishing turtle viewing maritime Gunung Kidul Yogyakarta 24 hours
Sadranan Beach	parking area gazebo home, stay prayer room, snorkeling photo spots, trash cans, toilets, food stalls, snorkeling sunbathing white sand, camping nautical Gunung Kidul Yogyakarta 24 hours
Krakal Beach	parking area home, stay prayer room, snorkeling photo spots, surfing trash cans, toilets, food stalls, swimming, snorkeling, surfing, camping, culinary restaurants, white sand, big waves, maritime Gunung Kidul Yogyakarta 24-hour
Ngrenahan Beach	parking area gazebo prayer room photo spots trash cans toilets coral hills cultural tourism swimming camping maritime Gunung Kidul Yogyakarta 24 hours

b. TFIDF Vectorizer

The corpus that has been obtained will be vectorized using the tf-idf vectorizer from the sklearn library. Term frequency (tf) in sklearn uses the formula term frequency adjusted for document length: $tf(t,d) = \frac{f(t,d)}{(\text{number of words in the document})}$, where term frequency

is expressed as the result of dividing the number of occurrences of words in the document with the total number of words contained in the document. Next, calculate the inverse document frequency (idf) using the log calculation formula using natural logarithm with base e 2.718281828459.... or it can be written $\log_e a$. The log calculation is carried out by dividing n , representing the total number of documents in a corpus plus one, and $df(t)$, representing the number of documents containing a certain term plus 1. Tf-idf vectorize in Sklearn also applies l2 normalization to the tf-idf weighting calculation. At this stage, it also produces a tf-idf matrix, which will be used to calculate cosine similarity. The results of tf-idf weighting on the corpus can be seen in Figure 6. with data visualization using the plotline library.

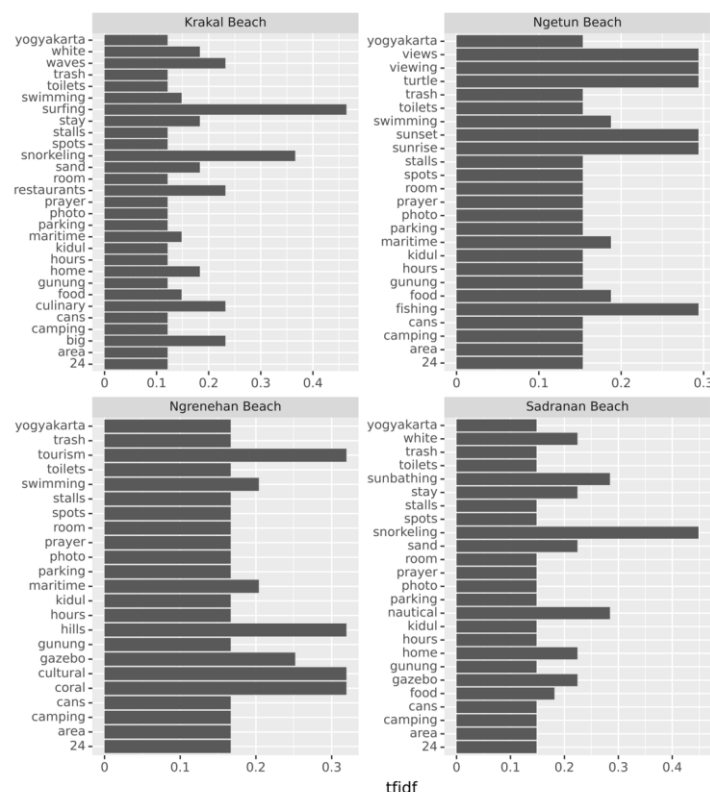


Figure 6. Visualization of TF-IDF Vectorizer

c. Recommend

After getting the weighting with the tf-idf vectorizer, at the recommendation stage the cosine similarity calculation is carried out on the tf-idf matrix. The cosine similarity calculation also uses the Sklearn library with a linear kernel, so it requires less time to execute. Cosine similarity is obtained by comparing two vectors between tf-idf matrices and then calculating to get a similarity score for each item or document so that the level of similarity between documents is obtained in the form of a cosine similarity matrix, which can be seen in Table 12.

Table 12. Cosine Similarity Matrix

	D1	D2	D3	D4
D1	1	0.398279	0.380666	0.485669
D2	0.398279	1	0.64306	0.45254
D3	0.380666	0.64306	1	0.383695

D4	0.485669	0.45254	0.383695	1
----	----------	---------	----------	---

From the cosine similarity matrix, it can be seen that Document 1 has the highest similarity to Document 4, with a similarity score of 0.485669. Apart from that, the level of similarity that has the same words is found in Document 2 and Document 3. The similarities between these documents can be seen by visualizing the data using the matplotlib and seaborn libraries in Figure 7.

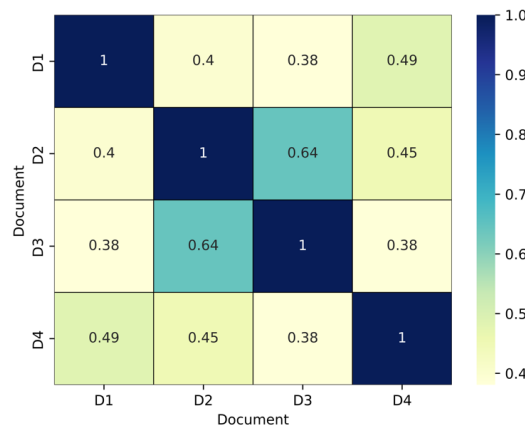


Figure 7. Cosine Similarity Matrix visualization

From the cosine similarity matrix, a pairwise similarity score is then carried out on the documents based on the index item, which is used as a reference in the recommendation process to obtain similarity values between documents. After getting the similarity score, the data is then sorted to get recommendation results based on the highest similarity value of the document. The sorting results of the pairwise similarity score can be seen in Table 13.

Table 13. Results of Content-Based Filtering Recommendations

Name	Metadata	Sim_Scores
Ngrenehan Beach	parking area gazebo prayer room photo spots trash cans toilets stalls coral hills cultural tourism swimming camping maritime Gunung Kidul Yogyakarta 24 hours	1
Ngetun Beach	parking area prayer room photo spots trash cans toilets food stalls swimming sunrise and sunset views camping fishing turtle viewing maritime Gunung Kidul Yogyakarta 24 hours	0.485669
Sadranan Beach	parking area gazebo home, stay prayer room, snorkeling photo spots, trash cans, toilets, food stalls, snorkeling sunbathing white sand, camping nautical Gunung Kidul Yogyakarta 24 hours	0.45254
Krakal Beach	parking area home, stay prayer room, snorkeling photo spots, surfing trash cans, toilets, food stalls, swimming, snorkeling, surfing, camping, culinary restaurants, white sand, big waves, maritime Gunung Kidul Yogyakarta 24-hour	0.383695

From the results of content-based filtering recommendations, it was found that items had a level of similarity that was close to the Ngrenehan Beach tourist attraction, namely Ngetun Beach with a similarity score of 0.485669, Sadranan Beach with a similarity score of 0.45254 and Krakal Beach with a similarity score of 0.383695.

CONCLUSION

The system in the Tour Guide and Travel Service application was built using a recommendation system with demographic filtering and content-based filtering methods. The application of a demographic filtering recommendation system is used to determine tour guide recommendations using a model that has three stages, including filtering, scoring and sorting. At the scoring stage, the system uses the Simple Additive Weighting method to provide a value as a tour guide rating. Apart from that, the implementation of a recommendation system using the demographic filtering method is used to recommend tourist attractions using a model that has two stages, namely training, and recommendation with document weighting using tf-idf vectorizer and determining the similarity between documents using cosine similarity. The use of demographic filtering and content-based filtering models provides personalized recommendation results so that the level of accuracy of the recommendation results becomes more personal and accurate.

REFERENCES

- Aggarwal, C. C. (2016). *Recommender systems*. Springer International Publishing.
- Aisha, D. (2022). *Online Store Recommendation System Using Collaborative Filtering and Content-Based Filtering Algorithms*. Maulana Malik Ibrahim State Islamic University.
- Ammar, M., Ahmad, UA, Dirgantoro, B., Saputra, RE, & Septiawan, RR (2022). Travelbuteng Application Development. Id For Tourism Promotion in Central Buton Regency (Android-Based Mobile Application module). *EProceedings of Engineering*, 9 (3).
- Aprianto, A. (2022). *TA: Application of the Content-Based Filtering Algorithm for Recommending Tourist Destinations in the Picnicker Application*. Dynamics University.
- Aprilian, LV, & Saputra, MHK (2020). *Learn the SAW method quickly*. Creative.
- Cahyadi, AD (2020). *Development Of A Tour Travel Service Information System And Tour Package Recommendation System In Yogyakarta Using The Content-Based Filtering Method*. Atma Jaya University, Yogyakarta.
- Falk, K. (2019). *Practical recommender systems*. Simon and Schuster.
- Gorakala, S. K., & Usuelli, M. (2015). *Building a recommendation system with R*. Packt Publishing Ltd.
- Hermawan, H., & Rosyid, H. (2021). Design And Development Of An Android Mobile Application For The Gresik District Tourist Location Information System Using The Item-Based Collaborative Filtering Method. *Indexia: Informatics and Computational Intelligent Journal*, 2 (1).
- Huda, AA, Fajarudin, R., & Hadinegoro, A. (2022). Content-based Filtering Recommendation System Using TF-IDF Vector Similarity for News Article Recommendations. *Building of Informatics, Technology, and Science (BITS)*, 4 (3), 1679–1686.
- Hutabarat, KI (2022). *Implementation of the K-Nearest Neighbors Algorithm in a Book Recommendation System Using Collaborative Filtering Techniques*. Medan Area University.
- Hyde, R. (2020). *Write Great Code* (volume 3). No Starch Press.
- Müller, A.C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc.
- Nanda, P. (2023). *Description of the Smoking Habits of Family Members in the Event of Acute Respiratory Infection (ARI) in Toddlers at the Putri Ayu Community Health Center*. Jambi University.
- Negre, E. (2015). *Information and recommender systems*. John Wiley & Sons.
- Ningrum, AS, Rustamaji, HC, & Fauziah, Y. (2019). Content-Based and Collaborative Filtering on Recommended Tourism Destinations in the Yogyakarta Region. *Telematics: Journal of Informatics and Information Technology*, 16 (1), 44–51.

- Pajankar, A., & Joshi, A. (2022). *Hands-on Machine Learning with Python: Implementing Neural Network Solutions with Scikit-learn and PyTorch*. Springer.
- Prasetyo, B., Haryanto, H., Astuti, S., Astuti, EZ, & Rahayu, Y. (2019). Implementation of the Item-Based Collaborative Filtering Method in Providing Recommendations to Potential Buyers of Smartphone Accessories. *Journal of Informatics Exploration*, 9 (1), 17–27.
- Pratama, R., Herdiansyah, MI, Syamsuar, D., & Syazili, A. (2023). Predicting Insurance Company Customer Retention Using Machine Learning. *Sisfokom Journal (Information and Computer Systems)*, 12 (1), 96–104.
- Pulungan, SM, Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2023). Entity-Relationship Diagram Technical Analysis in Database Design. *Journal of Management Economics and Business (JEMB)*, 1 (2), 98–102.
- Ramadhan, F., & Purwandari, N. (2018). Web-Based Sales Information System at PT. Mustika Teak. *Journal of Science And Technology*, 5 (1), 43–57.
- Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, sci-kit-learn, and TensorFlow 2*. Packt Publishing Ltd.
- Syahreza, MF, & Tanjung, IS (2018). Motives and patterns of use of Instagram social media among students of the UNIMED Economic Education Study Program. *Journal of Interaction: Journal of Communication Studies*, 2 (1), 61–84.
- Thanaki, J. (2017). *Python natural language processing*. Packt Publishing Ltd.
- Wijaya, AE, & Alfian, D. (2018). Laptop Recommendation System Using Collaborative Filtering and Content-Based Filtering. *Journal of Computech & Business*, 12 (1), 11–27.
- Yen, D. C. (2019). *The Information Systems Consultant's Handbook: Systems Analysis and Design*.
- Yoshua, I., & Bunyamin, H. (2021). Implementing a Music Recommendation System Using the Collaborative Filtering Method. *STRATEGY Journal-Maranatha Journal*, 3 (1), 1–16.



© 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).