

ECOTOURISM RECOMMENDATIONS BASED ON SENTIMENTS USING SKYLINE QUERY AND APACHE-SPARK

Nouval Trezandy Lapatta

Faculty of Teacher Training and Education, Tadulako University, Indonesia

Email: nouval@untad.ac.id

ARTICLE INFO

ABSTRACT

Received : 21 April 2022

Revision : 02 May 2022

Received : 15 May 2022

Keywords:

Ecotourism;
recommender system;
skyline query;
sentiments
spark

The selection of an ecotourism destination is a challenging service in an online transaction. The process must consider personal considerations, such as costs or distance and interesting eco-points like specific sceneries or the rare and unique picturesque landscapes. Only a few tourists have such required information for any particular local resources. A proposed recommender system is a solution for tourists to get advice on appropriate ecotourism destinations based on sentiments according to their preferences. This work proposed the skyline query method based on the Skyline Sort Filter algorithm in the Apache Spark cluster computing framework to build recommendations. The sentiment analysis process using the SentiStrength algorithm obtain an accuracy of 78.3% and F-arithmetic of 84.5%. These results indicate the proposed recommender system can detect positive responses from visitors to ensure best ecotourism recommendations with positive sentiments for tourist. Apache Spark with three computer nodes has 213.7 times faster execution time on correlated data, 240 times faster on independent data, and 288.1 times faster on anti-correlated data than a single computing method.

Introduction

Recommender Systems are software tools and techniques that provide items suggestions that will be useful to users (Ricci, Rokach, & Shapira, 2011). Suggestions given are intended to help users in various decision-making processes (Mahmood & Ricci, 2009), such as choosing news, music (Sunitha & Adilakshmi, 2018), or tourist destinations (Gavalas, Konstantopoulos, Mastakas, & Pantziou, 2014). The growth of decision support systems and increasing data size lead researchers to seek new recommendation methods to efficiently retrieve useful insights from multi-dimensional datasets (Kalyvas & Tzouramanis, 2017). An efficient recommendation method can provide the best advice or recommendations regardless of user preferences. For example, a user wants to get the nearest tourist destinations advice

at a low cost in tourism. When there are no objects that meet these criteria, the recommender system must be able to provide other interesting alternative suggestions but still meet the previous evaluation criteria such as near and cheap.

Tourism activities have a variety of certain characteristics including preference of tourists in choosing a tourism destination. For example, in choosing an ecotourism object (Damanik & Weber, 2006), a user might consider the rare or unique flora and fauna, beautiful sceneries, ease of access, or available facilities. Various criteria considered and increasing size of data cause searching the database using the conventional method will require high computation and may not produce the expected results.

In recent years, the Skyline query method has become an important issue in database research to extract interesting

objects from multi-dimensional datasets. Skyline query processing applies in many applications that require multi-criteria decision making without using cumulative functions to determine the best results based on user preferences. The skyline operator (Borzsony, Kossmann, & Stocker, 2001) filters out a set of interesting objects based on evaluation criteria from a large objects dataset. Interesting objects are objects not dominated by other objects in data. An object is said not to be dominated by other objects if the value of an object is better on all criteria and better at least on one criteria (Djatna & Morimoto, 2009). Skyline query can be used to find the interesting ecotourism objects that are not dominated by other objects with certain characteristics.

Previous study developed a mobile ecotourism recommendation system using spatial data of ecotourism object, user's profile, and frequency of visits, but the data used is static or not dynamic (N. Rosmawarni, T. Djatna, 2013). In addition, tourism recommendation modelling by measuring the similarity between a user's profile and characteristics of a tourism object extracted from the tourism object's social media account (Khotimah, Djatna, & Nurhadryani, 2014). However, the characteristics do not represent the tourism object because the social media account also posts irrelevant to tourism object and its activities. This study attempts to answer the lack of previous research.

A dynamic and relevant ecotourism recommendation requires representative and updated data on the ecotourism object. The tourism sites like TripAdvisor, besides general preferences such as distance or cost in the recommendation method, also use input from tourists, such as ratings and comments. This method has a weakness in that someone can give a good rating with a bad comment or vice versa. Sentiment analysis can be applied to determine the sentiment of visitor comments precisely, whether positive, negative, or neutral (Medhat, Hassan, & Korashy, 2014). Then the sentiment score will be combined with the rating given. Through this sentiment analysis process, tourists are expected to get ecotourism recommendations with the best rate and positive responses.

The number of preferences considered, a large amount of data, and the sentiment

analysis process applied can cause high complexity. Thus, computational process to produce these recommendations cannot be done using conventional methods or running on a single computer. A solution is to implement Skyline query processing with a cluster computing method (Ramdani, Djatna, & Sukoco, 2018). Cluster computing can process the task given on multiple computers in parallel. This work uses Apache Spark as a cluster computing framework. The use of cluster computing methods through Apache Spark is expected to increase the speed of generating recommendations. Finally, this study aims to develop a Skyline query to generate ecotourism recommendations based on sentiment using Apache Spark. The proposed recommender system has been implemented in an Android mobile application.

Method

The tools used in this study consist of hardware for cluster architecture and mobile application testing devices, and software for developing a recommender system. This cluster uses four nodes of virtual machines from Google Cloud Platform (GCP) with their specifications as shown in Table 1. One computer acts as a primary node that sends tasks through the cluster manager to be processed simultaneously on the other three computers as the executor called the worker node.

Table 1
Cluster architecture specifications

Component	Configuration	Version
Processor	2 vCPU	Broadwell
Memory	8 GB	DDR3
Storage media	500 GB	HDD
Operating system	Ubuntu 18.04 LTS	1.4-ubuntu18

The minimum specifications of mobile devices for implementation testing are shown in Table 2. Moreover, the software used in developing this recommendation system is shown in Table 3. Ecotourism data was collected from the Indonesia Ministry of Environment and Forestry, TripAdvisor site, and Google Maps API.

Table 2
Minimum specification of mobile devices

Component	Configuration
Memory	2 GB
System Development Kit (SDK)	API 24 Android 7.0 Nougat
Screen resolution	1280 x 800

Table 3
Software used

Software	Version	Function
Anaconda	5.2.0	Python packet management and environment
Android Studio	3.4.1	Integrated Development Environment (IDE) for Android development
Apache Spark	2.4.2	Cluster-computing framework
Jupyter Notebook	4.4.0	Editor for Python application development
Kotlin	1.3.31	Programming language for Android development
Python	3.7.3	Application Programming Interface (API) language used in Apache Spark

The method consists of 2 stages, pre-processing and recommendation method development processed within the Apache Spark cluster framework, as shown in Figure 1.

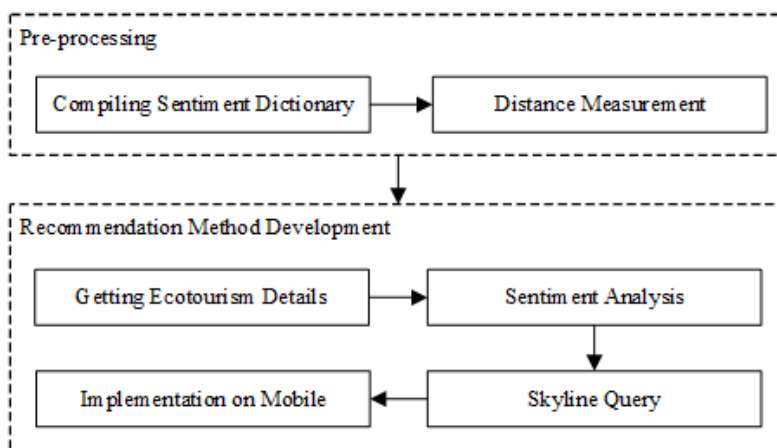


Figure 1. The flow of research stages

a. Compiling sentiment dictionary

This study uses sentiment analysis with the lexicon method based on dictionaries through the SentiStrength algorithm. The sentiment dictionary used is an adaptation and translation of English

words into Indonesian (Liu, Hu, & Cheng, 2005). Table 4 shows the dictionaries of sentiments used and examples of Indonesian words, and the sentiment scores.

Table 4
Sentiment dictionaries

Dictionary	Example words	Sentiment score
Booster	<i>sangat</i>	2
Emoticon	<i>:(</i>	-3
Idiom	<i>besar kepala</i>	-4
Negation	<i>bukan</i>	-1
Negative	<i>kotor</i>	-3
Positive	<i>senang</i>	4
Question	<i>apakah</i>	0

b. Distance Measurement

Distance measurement is done by comparing the ecotourism location to the user's current location or another specified location using the Haversine formula (Equation 1). The result of this process is the distance attribute, which will be used as a preference for the recommendation system.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

(1)

Where φ is latitude, λ is longitude, and r is the radius of Earth.

c. Getting Ecotourism Details

Ecotourism location consists of latitude and longitude coordinates. By using these coordinates and the Application Programming Interface (API) in Table 5, detailed information about the object of ecotourism will be obtained in Javascript Object Notation (JSON) format. It will be used in the sentiment analysis and dominance test stage.

Table 5
API format used

API	Endpoint	Output
Distance Matrix	https://maps.googleapis.com/maps/api/distancematrix/parameter	Distance and route to the object
Open Weather	https://api.openweathermap.org/data/version/weather?parameter	The weather around the object
Place Details	https://maps.googleapis.com/maps/api/place/details/parameter	Detailed information about the object

d. Sentiment Analysis

An information obtained in the previous stage is a list of comments given by tourists about the ecotourism object. Based on the results, sentiment analysis is applied to determine tourist sentiment for the ecotourism object. The algorithm used for sentiment analysis is SentiStrength (Thelwall, Buckley, Paltoglou, Cai, & Kappas, 2010). SentiStrength is an algorithm with a lexicon-based classification that uses rules and additional linguistic information (non-lexical) to measure the sentiment power of short text in English (Wahid & Azhari, 2016).

SentiStrength uses positive and negative scales. This is based on psychological research, which states that humans can independently feel positive and negative emotions simultaneously to a certain extent (Norman et al., 2011). SentiStrength will produce positive and negative values in 1 to 5. Value 1 indicates sentence lacks of positive or negative sentiment, while value 5 indicates sentence has a very positive or negative sentiment (Thelwall, Buckley, & Paltoglou, 2012). Based on the

sentiment score, the sentiment class of a comment text will be decided by comparing the highest positive and the highest negative score with the following rules:

- 1) If positive > negative, then positive sentiment.
- 2) If positive < negative, then negative sentiment.
- 3) If positive = negative, then neutral sentiment.

Furthermore, the class score is obtained from the difference between the maximum positive and maximum negative score. Then summarize between class score and rating score. The highest value obtained will be the value of sentiment attribute and used as a preference to rank recommended ecotourism objects at the next stage.

e. Skyline Query

Recommended objects are ranked using the skyline query method in the Apache Spark cluster framework. The skyline query algorithm used is Sort Filter Skyline (SFS) (Chomicki, Godfrey, Gryz, & Liang, 2003). SFS is a development of the predecessor algorithm, Block Nested Loop (BNL). Like the naive nested-loop algorithm,

BNL repeatedly reads the set of tuples and eliminates objects by finding other objects in the dataset that dominate them. Its performance is susceptible to the number of dimensions and the underlying data distribution.

SFS improves BNL performance by pre-sorting the input dataset in ascending order according to a monotone preference function, such as the sum of values of an object on all dimensions or optimized as entropy. Presorting enforces that an object p dominating another object q will be visited before q . This reduces the number of pairwise comparisons between objects and ensures the progressive behaviour of SFS. The fewer dominance tests performed, SFS is significantly more efficient in its computation. The following are the attributes of each ecotourism object data used to generate recommendations.

- 1) Flora: rare plants, medicines, forests.
- 2) Fauna: endemic or rare animals.
- 3) Sceneries: a beautiful spot for photography needs.
- 4) Facilities: number of tourist facilities or playgrounds.
- 5) Access: access to ecotourism locations.
- 6) Rate: ecotourism rate by tourists on Google Maps.
- 7) Distance: the value of the distance between tourists and the ecotourism location. This value is obtained from the distance measurement process using Equation 1.
- 8) Sentiment: a sentiment score from tourist comments. This score is obtained from the sentiment analysis process using the SentiStrength algorithm.

This study developed method of Skyline query by implementing multi-level Skyline queries (Kodama, Iijima, Guo, & Ishikawa, 2009). The algorithm works well, but the problem is resulting skyline may consist of a small number of objects. A user who wants to compare several destinations would not be satisfied by such a result. If the number of skyline objects is not greater or equal to the user's request, then search for the

next skyline object by removing skyline object that has been obtained from next candidate list.

Based on the object data of ecotourism along with all its attributes and preferences, then for each ecotourism object t , where $t = [1, 2, 3, 4 \dots n]$ is carried out, the ranking process of object recommendations through the dominance test using the SFS algorithm with the following stages.

- 1) Presorting is based on the entropy value obtained from Equation 2.

$$E(t) = \sum_{i=1}^d \ln(t[a_i]+1)$$

Where $E(t)$ is the entropy value of object t , and $t[a_i]$ is the normalized value of an attribute of object t in the i -dimension.

- 2) The t object at the top of pre-sorted data (entropy 1) is the first skyline object.
- 3) For each subsequent t object, dominance tests is used with the current skyline object in a window (S).
- 4) If t is dominated by skyline object in S , delete t .
- 5) If the skyline object in S does not dominate t , save t as a new skyline object.
- 6) If the number of skyline objects generated is smaller than the user's preference, repeat the steps by removing the current skyline object as the next candidate.

f. Implementation of Mobile Applications

The recommender system workflow on the Android mobile application can be seen in Figure 2. The preference is input from the user, such as current location or distance from user. The ecotourism list is obtained from database, while detailed information is obtained from the Google Maps API. Then rank the recommended objects through the skyline query method within Apache Spark, and the results as the ecotourism recommendation will be displayed on the user's device.

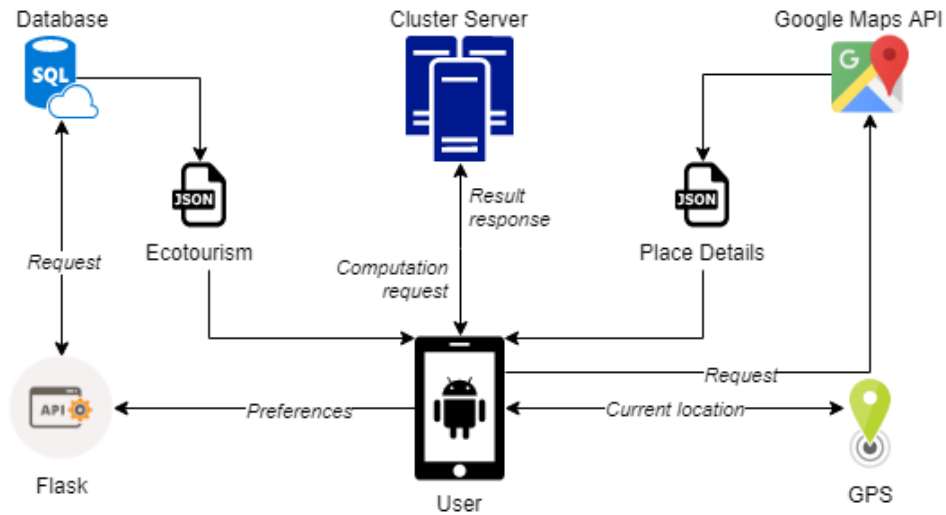


Figure 2. Implementation of Android application

g. Evaluation Scenario

Evaluation is important to ensure that the algorithms and methods are working correctly. In this work, the evaluation was divided into two: the evaluation of sentiment analysis using the SentiStrength algorithm and the evaluation of the increase in execution time using the cluster computing

methods through the Apache Spark framework compared to a single computing method. The sentiment analysis results were evaluated by calculating the accuracy and F-measure (F1) based on precision and recall (Junker, Hoch, & Dengel, 1999) measurements, as shown in Equations 3-6.

$$\text{Accuracy} = \frac{\sum \text{Correct predicted (True)}}{\sum \text{Total population}} \tag{3}$$

$$\text{Precision} = \frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}} \tag{4}$$

$$\text{Recall} = \frac{\sum \text{True positive}}{\sum \text{Condition positive}} \tag{5}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

Enhancement of the processing time (latency) obtained from cluster computing implementation is evaluated using the speedup factor (*S*), as shown in Equation 7.

$$S_{\text{latency}} = \frac{L_s}{L_c} \tag{7}$$

L_s is latency in single computing, and *L_c* is latency in cluster computing.

Results And Discussion

This section will explain the sentiment analysis results applied to visitor comments to get the most positive response. In addition, it will also discuss the development of recommendation methods using skyline queries and the enhancement of execution

time obtained from the Apache Spark cluster computing implementation compared to the single computing method.

A. Sentiment Analysis

Sentiment analysis is performed on visitor comments on the Google Maps

application using the SentiStrength algorithm to generate sentiment class, whether positive, negative, or neutral. This sentiment class is obtained by comparing the maximum positive and negative scores. At the same time, the class score is obtained from the difference between maximum positive and maximum negative scores. By summing class score the rating given by tourists, the sentiment

attribute value will be obtained and used to rank the recommended object using Skyline Query algorithm in the next stage.

Table 6 illustrates the sentiment classification results of Indonesian comment text using the SentiStrength algorithm, where the maximum positive score is 4 while the maximum negative score is 3. Then the sentiment class of text will be a positive sentiment.

Table 6
Example of sentiment classification results

Information	Value
Comment text	<i>Pemandangannya sih sangat bagus tapi tempatnya agak jauh :(</i>
Classified text	<i>pemandangannya sih sangat bagus [6] tapi tempatnya agak jauh [-1] :([-3]</i>
Maximum positive score	6
Maximum negative score	-3
Sentiment class	Positive
Class score	3

The sentiment classification results were evaluated using 368 tourist comment data from 78 ecotourism objects on the Google Maps application, as shown in Table 7. Evaluation scores obtained were 78.3%, precision 87.7%,

recall 81.5% and F- measure of 84.5%. These results indicate that the SentiStrength algorithm has been able to classify positive sentiments well, but negative sentiment comments still need better analysis in the pre-processing stage.

Table 7
Evaluation of sentiment classification results

Actual	Predicted			True	False	Total
	Positive	Neutral	Negative			
Positive	8	18	185	185	26	211
Neutral	5	92	29	92	34	126
	11	7	13	11	20	31
Total	24	117	227	288	80	368

B. Development of Recommendation Method

Based on available attributes and preferences given by users, ecotourism object recommendations are generated using the skyline query through the Sort Filter Skyline (SFS) algorithm with a multi-level skyline query method. This test is carried out using the preferences as follows:

- 1) For the distance attribute, minimal value is better.
- 2) For another attribute, maximal value is better.

- 3) The number of recommendations expected by the user (k) is 6, with the maximum distance being 200 km.

The result of the ranking process is shown in Table 8. Based on the preferences, the number of skyline objects generated in the first process (Level 1) is 4 objects. This number is less than the user's request. As a result, Level 2 Skyline query process is carried out by removing the skyline object generated in Level 1 from the next candidate list. Then Level 2 generates 4 skyline objects, the total number of skyline objects is 8. This number is more than the user's request, thus the Skyline

query process stops at the second level, and the results are delivered to the user.

Table 8
Result of multi-level skyline query with k=6

Level	Skyline object	Attributes*								Candidates	Time (ms)
		1	2	3	4	5	6	7**	8		
1	TN Pancar Mountain	2	3	3	3	3	5	1.5	8	15	0.998
	TN Ujung Kulon	3	3	3	3	3	5	153.8	4		
	SM Sawal Mountain	3	2	2	3	2	5	180.3	9		
	TN Halimun Salak Mountain	3	3	2	3	2	3	13.1	4		
2	SM Muara Angke	2	3	3	3	3	5	51.2	7	11	0.996
	TN Ciremai Mountain	3	3	3	3	3	5	187.3	1		
	TN Gede Pangrango Mountain	2	2	3	2	2	5	33.1	3		
	TWA Cimanggu	2	2	2	2	2	3	30.8	3		

* value before normalized, ** distance attribute (in km)

The process of ranking recommendation objects using the skyline query in the previous test has been successfully carried out, with the execution time still relatively fast. However, the data size increases and the number of preferences considered is also increasingly complex. This

study conducted a test using single computing with different data sizes, and distributions are correlated, independent, and anti-correlated. Figure 3 shows the execution time increases exponentially as the data size increases.

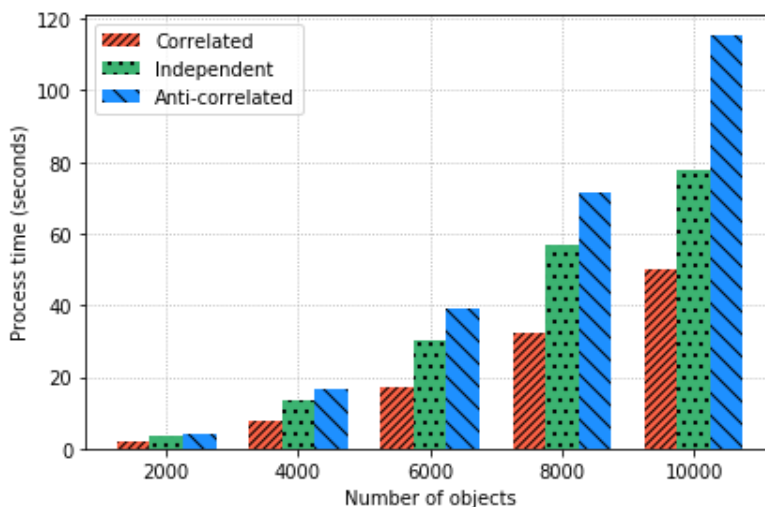


Figure 3. Single computing test result

The increasing size of the data over time and the need for fast access to information on mobile today cause a recommendation process using conventional or single computing methods is not ideal. One of the solutions is to use cluster computing methods. This study applies the cluster computing method through Apache Spark with architecture, as shown in Figure 4. One

computer acts as a primary node that sends tasks through the cluster manager to be processed simultaneously on other three computers as the executor called worker node. The use of cluster computing methods through the Apache Spark framework results in process time to generate recommendations significantly faster than using the single computing method.

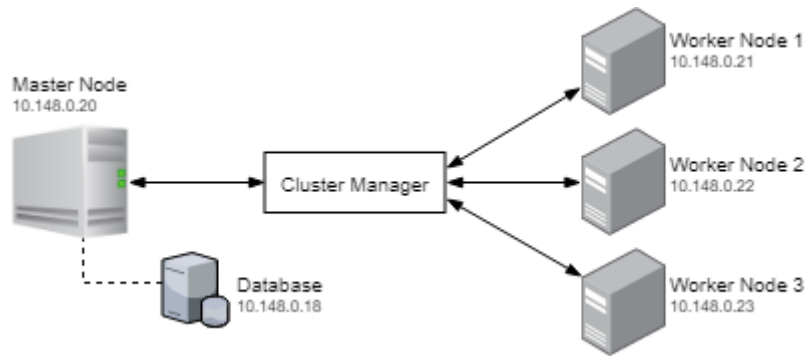


Figure 4. Cluster architecture using Apache Spark

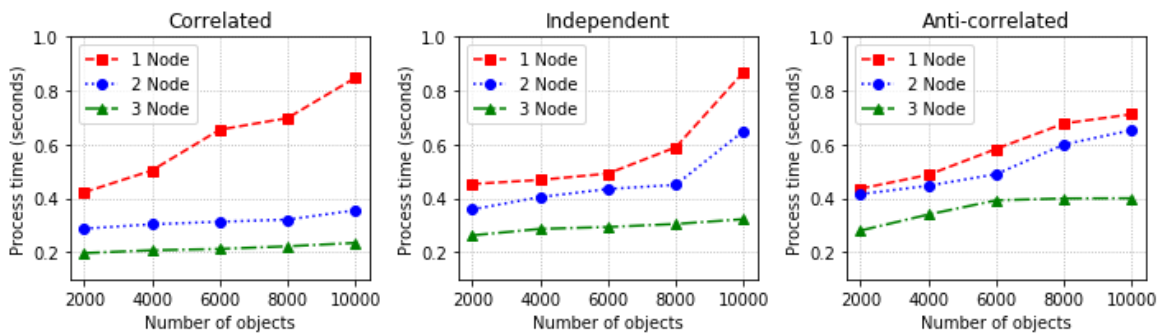


Figure 5. Results of the recommendation method testing on cluster computing

Figure 5 shows a testing result's graph of cluster computing method using different data distributions and number of nodes. The processing time is below 1 second consistently and increases as the number of nodes in the cluster increases. This happens because the Apache Spark framework implements the cluster computing method

where a task is processed by multiple computers simultaneously, make processing time is faster. Based on the previous evaluation scenario, Table 9 shows the speedup of processing time by implementing cluster computing methods. Value 4.7 means cluster computing processing time is 4.7 times faster than single computing method.

Table 9
Speedup results of cluster computing method

Number of objects	Correlated			Independent			Anti-correlated		
	Number of nodes								
	1	2	3	1	2	3	1	2	3
2000	4.7	6.8	10.0	7.8	9.8	13.4	9.2	9.6	14.3
4000	15.8	26.2	38.4	28.6	33.0	46.7	34.4	37.5	49.3
6000	25.8	53.9	79.5	61.1	69.1	102.3	67.2	80.1	99.9
8000	46.4	101.0	146.0	96.0	125.8	185.7	105.1	118.8	178.3
10000	59.1	140.8	213.7	89.2	119.2	240.0	161.9	176.5	288.1

Figure 6 shows a simulation of ecotourism recommender system implementation on a map. The red pin on the image shows user's current location obtained through GPS sensor in a mobile device. In comparison, several green pins show

surrounding ecotourism objects. Based on several ecotourism objects available and user preferences, the proposed recommender system will provide the best ecotourism destinations advice to the user with a positive response from visitors.

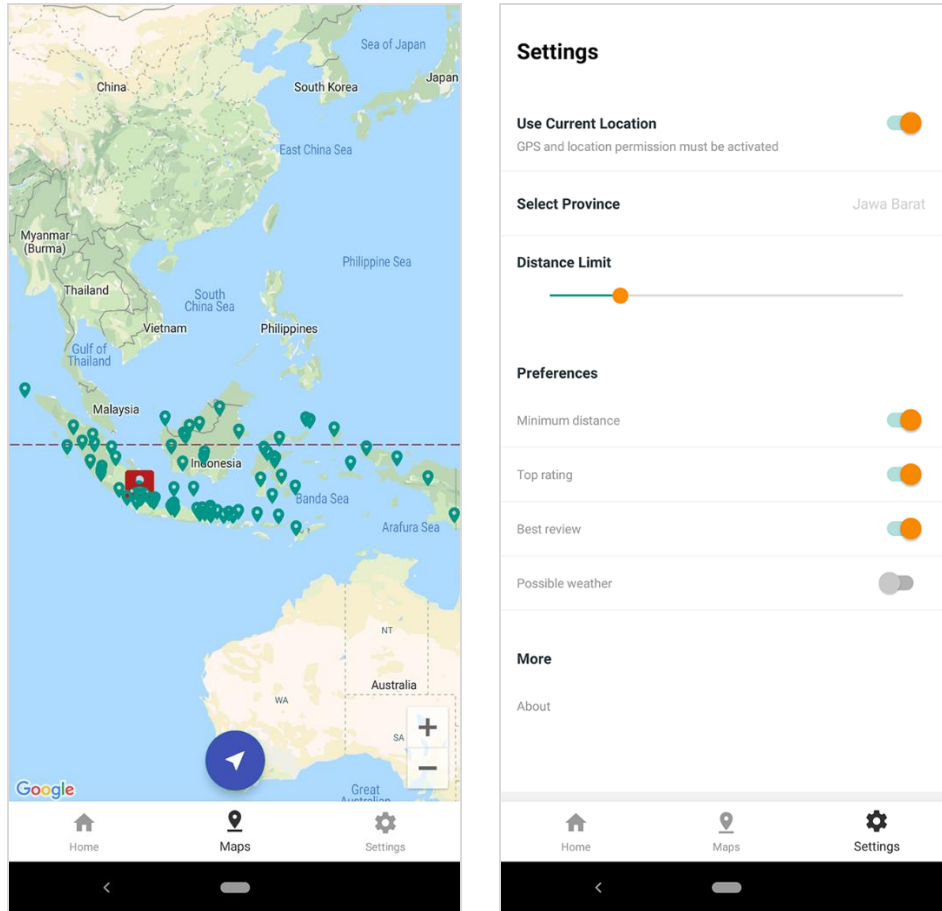


Figure 6. List of ecotourism objects and preference settings page

The results of ecotourism recommendations will be displayed on user's device's screen, as shown in Figure 7. From the list of recommendations given, users can choose to see detailed information about

ecotourism object such as location, contact number, the most positive comment labelled as the top review, and other supporting information.

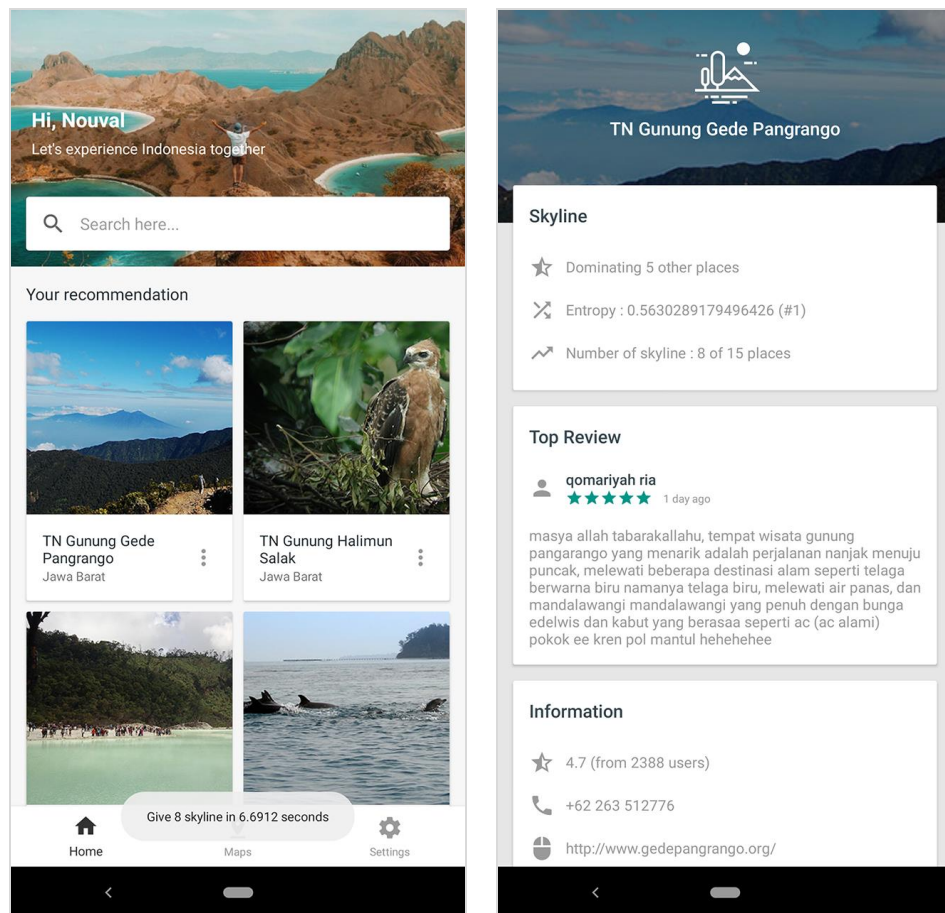


Figure 7. Recommendation interface for Android application

Conclusion

This study has developed Skyline query method and Apache Spark framework to generate ecotourism recommendations based on sentiment. The SentiStrength algorithm successfully classifies visitors' comments with positive sentiments by obtaining an accuracy score of 78.3% and F-arithmetic of 84.5%. This result means proposed recommender system has been able to provide ecotourism recommendations with visitors' positive responses.

The implementation of Apache Spark cluster computing method by using three nodes has succeeded in increasing the speed of recommended object ranking process compared to conventional method or single computing, which is 213.7 times faster in correlated data, 240 times faster in independent data, and 288.1 times faster in data anti-correlated.

Suggestions given for further research include the Skyline query method can be developed to provide advice on supporting tourism facilities based on ecotourism recommendations that were previously generated. For example, looking for the nearest hotel from a recommended ecotourism object is far from the crowds but close to a terminal or station, perform a better pre-process method to improve classification accuracy, especially for comments with negative sentiment, and generate preference attributes automatically based on the topic of visitor comments. For example, visitors comment that the facility is good or bad.

References

- Borzsony, S., Kossmann, D., & Stocker, K. (2001). The skyline operator. *Proceedings 17th International Conference on Data Engineering*, pp. 421–430. IEEE. [Google Scholar](#)
- Chomicki, J., Godfrey, P., Gryz, J., & Liang, D. (2003). Skyline with pre-sorting. *ICDE, 3*, 717–719. [Google Scholar](#)
- Damanik, J., & Weber, H. F. (2006). Ecotourism planning: from theory to application. *Yogyakarta. Andi Offset, 10*, 37–38. [Google Scholar](#)
- Djatna, T., & Morimoto, Y. (2009). Reviving Dominated Points in Skyline Query. *IJCSNS, 9*(1), 347. [Google Scholar](#)
- Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications, 39*, 319–333. [Scopus](#)
- Junker, M., Hoch, R., & Dengel, A. (1999). On evaluating document analysis components by recall, precision, and accuracy. *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, 713–716. IEEE. [Google Scholar](#)
- Kalyvas, C., & Tzouramanis, T. (2017). A survey of skyline query processing. *ArXiv Preprint ArXiv:1704.01788*. [Google Scholar](#)
- Khotimah, H., Djatna, T., & Nurhadryani, Y. (2014). Tourism recommendation based on vector space model using composite social media extraction. *2014 International Conference on Advanced Computer Science and Information System*, 303–308. IEEE. [Google Scholar](#)
- Kodama, K., Iijima, Y., Guo, X., & Ishikawa, Y. (2009). Skyline queries are based on user locations and preferences for making location-based recommendations. *Proceedings of the 2009 International Workshop on Location-Based Social Networks*, 9–16. [Google Scholar](#)
- Liu, B., Hu, M., & Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. *Proceedings of the 14th International Conference on World Wide Web*, 342–351. [Google Scholar](#)
- Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, 73–82. [Google Scholar](#)
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal, 5*(4), 1093–1113. [Scopus](#)
- N. Rosmawarni, T. Djatna, and Y. N. (2013). A mobile ecotourism recommendations system using Cars-context Aware Approaches. *Telecommunication Computing Electronics and Control, 11*(4), 845–852. [Google Scholar](#)
- Norman, G. J., Norris, C. J., Gollan, J., Ito, T. A., Hawkley, L. C., Larsen, J. T., ... Berntson, G. G. (2011). Current emotion research in psychophysiology: The neurobiology of evaluative bivalence. *Emotion Review, 3*(3), 349–359. [Google Scholar](#)
- Ramdani, A. L., Djatna, T., & Sukoco, H. (2018). Selecting user influence on twitter data using skyline query under mapreduce framework. *Telkonnika, 16*(3), 1416–1425. [Google Scholar](#)
- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1–35). Springer. [Google Scholar](#)
- Sunitha, M., & Adilakshmi, T. (2018). Music recommendation system with user-based and item-based collaborative filtering

technique. In *Networking communication and data knowledge engineering* (pp. 267–278). Springer. [Google Scholar](#)

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1), 163–173. [Google Scholar](#)

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment

strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12), 2544–2558. [Google Scholar](#)

Wahid, D. H., & Azhari, S. N. (2016). Peringkasan sentimen esktraktif di twitter menggunakan hybrid TF-IDF dan cosine similarity. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 10(2), 207–218. [Google Scholar](#)

Copyright holder:

Nouval Trezandy Lapatta (2022)

First publication right:

Journal of Social Science

This article is licensed under:

