Accredited by National Journal Accreditation (ARJUNA) Managed by
Ministry of Research, Technology, and Higher Education, Republic Indonesia with Second Grade (Peringkat 2, Sinta 2)
since year 2017 to 2021 according to the decree No. 10/E/KPT/2019.

Published online on the journal's webpage: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 5 No. 5 (2021) 977 - 983 ISSN Media Elektronik: 2580-0760

The Hybrid Recommender System of the Indonesian Online Market Products using IMDb weight rating and TF-IDF

Muhammad Zuhdi Fikri Johari[1], Arif Dwi Laksito[2]
[1]Faculty of Computer Science, Universitas Amikom Yogyakarta
[2]Faculty of Computer Science, Universitas Amikom Yogyakarta
[1]muhammad.jo@students.amikom.ac.id, [2]arif.laksito@amikom.ac.id*

Abstract

Today, consumers are faced with an abundance of information on the internet; accordingly, it is hard for them to reach the vital information they need. One of the reasonable solutions in modern society is implementing information filtering. Some researchers implemented a recommender system as filtering to increase customers' experience in social media and e-commerce. This research focuses on the combination of two methods in the recommender system, that is, demographic and content-based filtering, commonly it is called hybrid filtering. In this research, item products are collected using the data crawling method from the big three e-commerce in Indonesia (Shopee, Tokopedia, and Bukalapak). This experiment has been implemented in the web application using the Flask framework to generate products' recommended items. This research employs the IMDb weight rating formula to get the best score lists and TF-IDF with Cosine similarity to create the similarity between products to produce related items.

Keywords: Recommender System, Indonesian Online Marketplace, Hybrid Filtering.

1 Introduction

In the last decade, online shopping has significantly increased because technology made it more accessible. Furthermore, in the previous year, the conditions make people get trapped at home. The pandemic forces people to utilize e-commerce to fulfill the need. The online shop companies try to provide the service for the customers to raise sales, expand the market, and gain trust.

Moreover, it is hard for users to reach the valuable information they need. One of the reasonable solutions in modern society is implementing information filtering. An efficient technique is required to deal with an overload of information that floods online users every day. A recommender system has been used to produce a top-n e-commerce products recommendation using the K-Nearest neighbor algorithm. This research divided data into data train and data tests, and then a confusion matrix was applied to evaluate it [1]. Generally, recommender system methods can be classified as collaborative filtering, content-based filtering, demographic filtering, and knowledge-based filtering [2]. The combination of more than one method in a recommender system is frequently called hybrid filtering. Sometimes, collaborative filtering suffers from several conditions, such as sparse data ratings, scalability, and cold start problems [3]. Content-based filtering does not need rating data to generate items recommendation since it used the product descriptions' similarity.

On the other hand, Wijaya et al. [4] state that using a combination method of collaborative and content-based filtering could deliver good laptop products' recommendation items. This research implemented collaborative filtering using an adjusted-cosine similarity algorithm and content-based filtering using TF-IDF (term frequency-inverse document frequency) method. Various research of the hybrid recommender system in the education domain has been done. It could increase learning quality, satisfy the learner [5], and outperform other recommendation methods in terms of quality and accuracy of recommendations [6]. In addition, The hybrid technique is practiced to create recommendation items in the e-tourism domain [7][8].

The main requirement in the recommender system is the data. In the text mining area, the method commonly used to collect the data is a crawling method. Suharno [9] used the social media crawling technique for his research in decision making. While other research implemented a crawling technique for automatic data collection using Twitter's Developer API [10][11].

Accepted by Editor: 11-09-2021 | Final Revision: 24-10-2021 | Online Publication: 31-10-2021

Therefore, this research focus on applying a recommender system in the Indonesian online market(Shopee, Tokopedia, and Bukalapak) to identify the right choice for many e-commerce products. The crawling approached has been adapted to this research to collect the products from each online market. This experiment is implemented in the web application using the Flask framework to generate products' recommended items. This research also applies the hybrid method, which combines demographic filtering using the IMDb weight rating formula and the content-based filtering using TF-IDF with Cosine similarity to create the similarity between products to produce related items.

2 Research Method

The initial of this study is collecting product data using the crawling method. The next step is pre-processing data, and there are two phases. Firstly, cleaning the data and equating the data type. Secondly, text processing, such as tokenizing process, case folding, and removing punctuation and stopwords. Furthermore, this research evolves the recommender system model for forming the product ranking and generating similar products. Designing and implementing a web application has been done after that. Finally, This research evaluates the web application to ensure that the result of recommender items is equal to the model—the diagram proses as shown in figure 1.

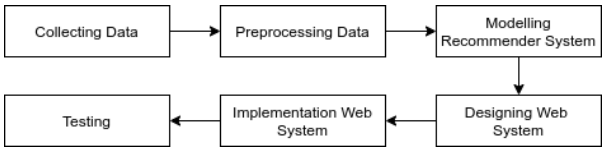

Figure 1. Research flow overview

2.1. Data Crawling

Data crawling is a method in the text mining or information retrieval research area that aims to collect from database or web page, and commonly it is called a web crawler [10]. Data collection using the crawling method starts by visiting a list of URLs called seeds. Data crawling tools, we used Selenium web driver, will then identify the contents on the web pages and automatically store them in the media storage [9]. The data crawling method has been implemented by some companies that offer information services to users. Accordingly, those companies can identify business opportunities, improve quality and quantity for their users.

2.1.1. Selenium WebDriver

Selenium WebDriver is one of the tools or libraries developed by the Selenium Project that enables developers, especially for Quality Assurance, to simulate running web applications automatically [12].

2.1.2. Data Visualization

Data visualization has an informative impress on the creation and study of visual representations of data. Utilizing attractive graphs can provide insight from a dataset rather than simply displaying it in tabular form [13]. In Business Intelligent (BI), data visualization is practiced to present the results of a set of methods, processes, architectures, applications, technology, etc., which are collected by transforming raw data into useful information to provide effective strategies to support decision making, see Figure 2.

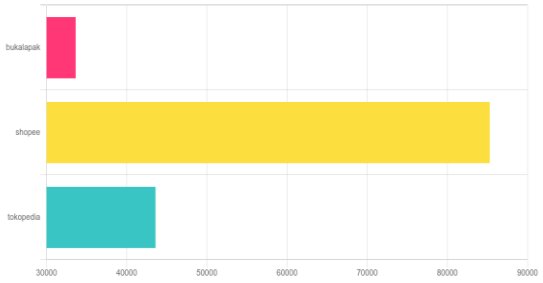

Figure 2. Graph data visualization

2.2. Preprocessing data

In the preprocessing stage, the results of csv file extraction are cleaned to keep the data consistent and there are no conflicts during the modeling process. In this study, data cleaning was carried out before or at the same time as the process of inputting or importing data into the database. The data cleaning process includes several things, such as updating the format or data type of the dataset and adding a default value in the missing data in the dataset.

2.3. Modelling E-commerce Recommender system

The recommendation system is analyzing and studies the behavior and characteristics of the users of an application. Hence, it can recommend a product registered in the system. The recommendation system's research objective is to help users determine which products are following the user's wants and needs. The recommendation system must meet four criteria, including relevance, novelty, serendipity, and variety [14]. E-commerce can be defined as using the internet network as a supporting mechanism for transactions such as buying, selling, and exchanging services and information digitally. E-commerce has the advantage in the efficiency and effectiveness of activities that can increase its users' goals and satisfaction. Accordingly, an e-commerce-based recommendation system is built to increase the sales of a product and increase service users. The e-commerce recommendation system processes information from e-commerce websites and analyzes it based on existing transaction products [15].

DOI: https://doi.org/10.29207/resti.v5i5.3486

2.2.1. Demographic filtering

Demographic filtering is a filtering method of the recommendation system that applies item recommendations according to the calculation of demographic data, such as age, gender, education, etc. This method is moderately simpler than the collaborative and content-based filtering methods because it does not require user ratings [16].

Table 1. Demographic data overview

Product name	Rate	Review
Laptop A1	4.8	1900
Laptop A2	4.7	880
Laptop A3	5.0	500

One of the calculation formulas that can be used for filtering demographic data is the IMDb weighted rating. IMDb Weighted rating developed by the Internet Movie Database (IMDb), a data and information provider company linked to films, television shows, video games, etc., including cast data, biographies, and production crews [17].

$$WR = \left(\frac{v}{v+m}\right) * R + \left(\frac{m}{v+m}\right) * C \quad (1)$$

It is shown that R is the average of the film ratings, v is the number of votes from the film, m is the minimum threshold for data criteria that has been used, and C is the average of the film's total votes.

2.2.2. Content-based filtering

Generally, the content-based method in the recommender system estimates the items for users based on the utilities assigned by the user to a similar item. For instance, in a movie recommender system, the movies that the user has given highly rating in the past will be utilized to a content-based method to recognize the commonalities among them [18]. The descriptive attributes of items are useful to make recommendations in the content-based method. The term “content” refers to these descriptions. The advantage of the content-based technique is it can make recommendations for new items meanwhile sufficient rating data are not available [19].

2.2.3. TF-IDF Weighting

The TF-IDF is a compound of two terms: Term Frequency (TF) and Inverse document frequency (IDF). The term TF is applied to calculate that how many times a term is present in a document. Any terms may come up more frequently in large documents than small documents. IDF is used to calculate how many terms appear in the entire document. IDF performs calculations for terms with the least frequency of appearance given a higher value, while more frequent appearances are lower.

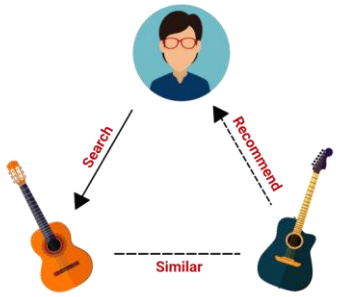

Figure 3. Content-based filtering overview

Consequently, TF-IDF will carry out a weighting process for each item to get important values or keywords in a set of words [20]. Then the TF-IDF weight is defined as.

$$w_{(x,y)} = tf_{(x,y)} * log\left(\frac{N}{df_x} - 1\right) \quad (2)$$

As stated earlier, tf is the frequency of terms (x) in a document (y), N is the total number of existing documents, df is the total number of documents containing a term.

2.2.4. Cosine Similarity

One of the methods is generally used to compare the similarity of documents or give a ranking of documents concerning a given vector of query words is cosine similarity. It can be said that the smaller the angle

between two vectors means the greater the match of them. On the other hand, when two vectors are at 90 degrees to each other, they have no match [21].

$$(t_a, t_b) = \frac{\sum(t_a.t_b)}{\sqrt{\sum t_a.\sum t_b}} \quad (3)$$

Where t_a is a term as a key and t_b is the term comparison.

2.4. Designing and Implementation Web system

A Web system implementation refers to integrating machine learning models to the production. In other words, exposing the machine learning models as an API endpoint provides the requests to applications.

This model can be implemented directly as a stand-alone predictor that provides output based on processing algorithms or as a dynamic model that provides requests in real-time [22].

2.5. Unit Testing

Unit testing is a simple test of the development process applied to a subset of code blocks. Unit testing includes easy tests and can also speed up the development process to improve developer confidence during code correction. Caching and debugging performed during unit testing are more accessible and cheaper [23].

DOI: https://doi.org/10.29207/resti.v5i5.3486

3. Result and Discussion

This section discusses the hybrid filtering recommendation system modeling results and their application to the web preview system as data visualization.

3.1. Demographic filtering using IMDb Weighted rating

The chart of a process calculating demographic filtering is shown in figure 4. Firstly, the data is retrieved from the database using a select query.

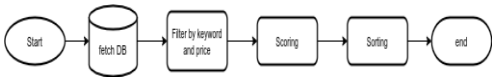

Figure 4. Flowchart Demographic filtering

The next step is filtering the data utilizing keywords and price. To get the expected score, the calculation is taken out using a predetermined formula. Finally, It is sorting data ordered from high score to the lowest. In the web application, demographic filtering possible to receive input from users.

At this stage, the IMDb Weighted rating, which was formerly used for the calculation of the top-n ranking of films by IMDb, is used to calculate the top-n products from the Indonesian online marketplace dataset. An example of practicing the IMDb weighted rating formula is shown in the program listing below. The calculation is taken using the rating column and product review column and using a quantile 0.9 to determine the minimum review limit as the product criteria to be recommended. The results of calculations using the IMDb weighted rating formula are in Figure 5, and there is a new column called score, which represents the results of product calculation.

IMDb Weighted Rating

```
# imdb weighted rating scoring function
def mp_score(df, q=0.9):
    df = df.copy()
    m = df.review.quantile(q)
    C = (df.rate * df.review).sum() /
        df.review.sum()
    df = df[df.review >= m]
    df["score"]    =    df.apply(lambda    x:
             (x.rate * x.review + C*m) /
             (x.review + m),
             axis=1)
    return df
# define scoring filter variable
keyword = "ssd 240gb"
prices = 100000
topk = 20
# get data based on filter
df = df[(df.keywords == keyword) &
        df.price.gt(prices)]
# call the function
df = mp_score(df)
# sort result data from highest score
rcm = df.loc[:, ("name",
                 "review",
                 "rate",
                 "score")]
rcm = rcm.sort_values(
    by=["score"],
    ascending=False)
```

3.2. Content-based filtering

Similarly, the chart of a process recommendation using content-based filtering is shown in figure 6. First of all, It is the same as demographic filtering by retrieving data from the database. It continued by text processing: case folding, tokenizing, and filtering by removing punctuation and stop words. Then the training stage uses the TF-IDF formula.

	name	review	rate	score
1190	SSD Midasforce Super Lightning [240GB]	913	5.0	4.970185
119	SSD WD Green 240GB 3D NAND SATA 2.5" 7mm Garan...	603	5.0	4.960425
1901	SSD 240GB Midas Force Super Lightning	380	5.0	4.948236
267	SSD Pioneer 240GB - SSD 240GB 2.5" SATA III	1500	4.9	4.898331
241	Western Digital WD Green SSD 240GB 2.5 SATA	1500	4.9	4.898331

Figure 5. Score result of IMDb Weighted rating

To determine the similarity between documents or content, a search is carried out using the cosine similarity algorithm based on the document index. For the results to be even better, filtering the data column is carried out based on product keywords.

In the recommendation process of content-based filtering, it conducts two steps to obtain products recommendation. Firstly, the word weighting using TF-IDF, and secondly, cosine similarity is used as a search for product closeness by utilizing the similarity of angles between documents.

3.2.1. TF-IDF Weighting

Table 2 describes dummy data as an example. Weighting is performed on each word, with the results in table 3 in the form of words that often appear in each document having a relatively lower score, even 0, than words that rarely appear in each document.

Table 2. Demographic data overview

Doc	Term
Q	original baterai charger laptop
D1	adaptor charger laptop
D2	mouse laptop usb, free windows 10 pro
D3	laptop dual baterai bonus mouse!
D4	ssd cocok untuk laptop

It can be defined that the smallest score (or zero) makes the word have no meaning. It is called stopwords, or even there is no word at all. In comparison, words with a high score are interpreted as meaningful words or called keywords from the document.

DOI: https://doi.org/10.29207/resti.v5i5.3486

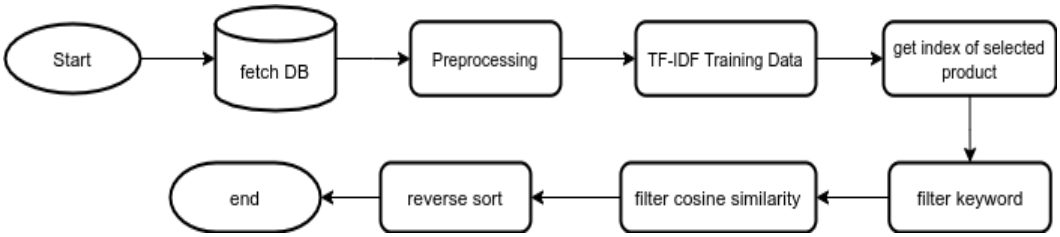

Figure 6. Flowchart Content-based filtering

Table 3. Result of TF-IDF Weighting

Term	TF					DF	IDF	TF-IDF				
	d1	d2	d3	d4	d5			d1	d2	d3	d4	d5
10	0	0	1	0	0	1	0.698	0	0	0.698	0	0
adaptor	0	1	0	0	0	1	0.698	0	0.698	0	0	0
baterai	1	0	0	1	0	2	0.397	0.397	0	0	0.397	0
bonus	0	0	0	1	0	1	0.698	0	0	0	0.698	0
charger	1	1	0	0	0	2	0.397	0.397	0.397	0	0	0
cocok	0	0	0	0	1	1	0.698	0	0	0	0	0.698
dual	0	0	0	1	0	1	0.698	0	0	0	0.698	0
free	0	0	1	0	0	1	0.698	0	0	0.698	0	0
laptop	1	1	1	1	1	5	0.000	0	0	0	0	0
mouse	0	0	1	1	0	2	0.397	0	0	0.397	0.397	0
original	1	0	0	0	0	1	0.698	0.698	0	0	0	0
pro	0	0	1	0	0	1	0.698	0	0	0.698	0	0
ssd	0	0	0	0	1	1	0.698	0	0	0	0	0.698
usb	0	0	1	0	0	1	0.698	0	0	0.698	0	0
windows	0	0	1	0	0	1	0.698	0	0	0.698	0	0

3.2.2. Cosine Similarity

In table 4, the same dummy data is used with the word weighting on the previous TF-IDF. For example, utilization of the cosine similarity algorithm is allocated in the Q document as the document key. Then each document compared using the weight score that has been carried out on TF-IDF. In this study, cosine similarity is applied to find closeness to the description content of the products contained in the dataset by providing alternative products that are similar to the description content chosen by the user.

Table 4. Result of Cosine Similarity

Document	Cosine Similarity	Percentage
adaptor charger laptop	0.096575	9.66%
mouse laptop usb free windows 10 pro	0.000000	0.00%
laptop dual baterai bonus mouse	0.048288	4.83%
ssd cocok untuk laptop	0.000000	0.00%

3.3. Recommender system model in the web

Figure 7 shows an example of implementing a recommendation system modeling on a web system using the Flask micro-framework. The recommendation system model that has been built previously is bundled in a module so that the Flask app can access it. The Flask app accesses the module by importing it, then access the functions in the module, and create an instance of calling the RecommenderSystem class. The use of the module's function on the web page is done by declaring a new instance of the required function as in Figure 8.

```
from flask import Flask, render_template, request, redirect
import pymysql
from recommendersystem import RecommenderSystem

app = Flask(__name__)

con = pymysql.connect(host='localhost', user='username', password='password', db='mp_recomsys')
cursor = con.cursor(pymysql.cursors.DictCursor)

query = "SELECT * FROM product"
recsys = RecommenderSystem(data=query)
recsys.fit()
```

Figure 7. Implementation Recommender system module to Flask

The web application has been integrated with the recommender system module as a result of the recommendation preview. In Figure 9, a page called Recommendation is presented, which contains details of

DOI: https://doi.org/10.29207/resti.v5i5.3486

the demographic calculation results to describe a real-time online marketplace.

```
@app.route('/data', methods=['POST', 'GET'])
def data():

    keyword = ['thinkpad x230', 'ssd 240gb', 'minna no nihongo', 'ps4', 'gitar akustik']
    count = 1
    price_input = 0
    topk_input = 20

    if request.method == 'POST':
        keyword_select = request.form['keyword']

        if request.form['keyword'] == keyword[2]:
            price_input = 80000

        if request.form['keyword'] == keyword[4]:
            price_input = 20000

        if request.form['price']:
            price_input = int(request.form['price'])

        if request.form['topk']:
            topk_input = int(request.form['topk'])

        data_entry = recsys.recommend_demo(keyword=keyword_select, prices=price_input, topk=topk_input)
        return render_template('data.html', active_page='data', data_entry=data_entry, keywords=keyword, count=count)
    else:
        data_entry = recsys.recommend_demo(keyword="thinkpad x230")
        return render_template('data.html', active_page='data', data_entry=data_entry, keywords=keyword, count=count)
```

Figure 8. Implementation Recommender system instance to Web

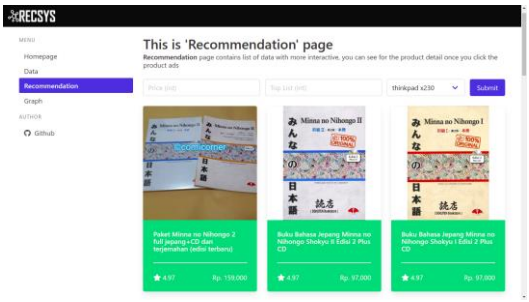

Figure 9. Demographic filtering implementation

Figure 10. Content-based filtering implemetation

Moreover, figure 10 presents the results of the content-based filtering calculation. A snippet of the Product Details page is revealed in information related to the previously selected product on the Recommendation page. There are generated products at the bottom of the page based on the search for product similarity displayed as product alternative recommendations

3.4. Evaluation Scenario

Finally, the evaluation of the web preview system is conducted using unit testing tools. Unit testing is done by examining the pages' navigation and functionality using automated testing methods on the web preview system to find out whether the web system can run and present the data properly or not. There are 6 units that have been tested as follows in table 5. It is clear that the page's tests' results can display data as expected in figure 11.

```
> python3 test.py -v
test_datapage (__main__.Testing) ... ok
test_detailpage (__main__.Testing) ... ok
test_graphpage (__main__.Testing) ... ok
test_homepage (__main__.Testing) ... ok
test_navigation (__main__.Testing) ... ok
test_recommendationpage (__main__.Testing) ... ok

----------------------------------------------------------------
Ran 6 tests in 646.290s

OK
```

Figure 11. Unit-testing evaluation

Table 5. Detail Unit-testing for Web

Unit	Detail testing
Test_datapage	Validation of whether the page can show product list based on the result of demographic filtering and user additional filter if needed
Test_detailpage	Validation of whether the user can access product detail from the recommendation page to show the detail of product and recommendation alternative based on the result of content-based filtering
Test_graphpage	Validation of whether the page can display calculation graph based on counted all product data and user can change different keyword
Test_homepage	Validation of whether the page can show an opening message with some explanation for the application
Test_navigation	Validation of whether each page can show as expected via the sidebar navigation button
Test_recommendationpage	Validation of whether page successfully loads data based on demographic filtering also loads images from the server, the user can add additional filters, and each product displayed can do trigger access to the details page

4. Conclusion

To conclude the research that has been done, demographic filtering using the IMDb weighted rating formula, which was originally used to determine the best top-n films, can also be applied to the calculation of the best top-n products in the Indonesian online marketplace. In content-based filtering, word weighting using TF-IDF and document similarity search with cosine similarity is proven to be applied to search for alternative product recommendations that have similar content with the product chosen by the user.

Future research is preferable to add more data variations in keywords so that the resulting recommendations are more varied. Another strategy assigns text preprocessing, such as lemmatization and stemming, so the cleaning process will be even better. If the recommendation system is to be implemented for mass users, it is better to stream data via API rather than request directly from the backend so that the program is relatively more flexible and scalable.

DOI: https://doi.org/10.29207/resti.v5i5.3486

Acknowledgement

Thank you to the Research and Community Service Department of Universitas Amikom Yogyakarta who provided funding through The Internal Research scheme.

References

[1] C. S. D. Prasetya, "Sistem Rekomendasi Pada E-Commerce Menggunakan K-Nearest Neighbor," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, p. 194, 2017.

[2] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intell. Data Anal.*, vol. 21, no. 6, pp. 1487–1524, 2017.

[3] X. Zhao, "A study on E-commerce recommender system based on big data," *2019 IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2019*, no. 1, pp. 222–226, 2019.

[4] A. E. Wijaya and D. Alfian, "Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering Dan Content-Based Filtering," *J. Comput. Bisnis*, vol. 12, no. 1, pp. 11–27, 2018.

[5] O. Bourkoukou and E. El Bachari, "Toward a hybrid recommender system for e-learning personalization based on data mining techniques," *Int. J. Informatics Vis.*, vol. 2, no. 4, pp. 271–278, 2018.

[6] J. K. Tarus, Z. Niu, and D. Kalui, "A hybrid recommender system for e-learning based on context awareness and sequential pattern mining," *Soft Comput.*, vol. 22, no. 8, pp. 2449–2461, 2018.

[7] M. E. B. H. Kbaier, H. Masri, and S. Krichen, "A personalized hybrid tourism recommender system," *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2017-Octob, pp. 244–250, 2018.

[8] R. Logesh and V. Subramaniyaswamy, *Exploring Hybrid Recommender Systems for Personalized Travel Applications*, vol. 2018-Janua, no. July. Springer, Singapore, 2018.

[9] F. A. Suharno and L. Listiyoko, "Aplikasi Berbasis Web dengan Metode Crawling sebagai Cara Pengumpulan Data untuk Mengambil Keputusan," *Semin. Nas. Rekayasa Teknol. Inf.*, no. November, pp. 105–109, 2018.

[10] J. Eka Sembodo, E. Budi Setiawan, and Z. Abdurahman Baizal, "Data Crawling Otomatis pada Twitter," in *Indonesian Symposium on Computing*, 2016, no. August, pp. 11–16.

[11] A. Dwi Laksito, Kusrini, H. Sismoro, F. Rahmawati, and M. Yusa, "A Comparison Study of Search Strategy on Collecting Twitter Data for Drug Adverse Reaction," *Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 356–360, 2018.

[12] S. Raghavendra, *Python Testing with Selenium*. Apress, 2021.

[13] A. Pajankar, *Practical Python Data Visualization*. Apress, Berkeley, CA, 2021.

[14] Hanafi, N. Suryana, and A. S. B. H. Basari, "An understanding and approach solution for cold start problem associated with recommender system: A literature review," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 9, pp. 2677–2695, 2018.

[15] S. Sfenrianto, M. H. Saragih, and B. Nugraha, "E-commerce recommender for usage bandwidth hotel," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 9, no. 1, pp. 227–233, 2018.

[16] M. Sridevi and R. Rajeswara Rao, "DECORS: A Simple and Efficient Demographic Collaborative Recommender System for Movie Recommendation," *Adv. Comput. Sci. Technol.*, vol. 10, no. 7, pp. 1969–1979, 2017.

[17] "IMDb | Help - Weighted Average Ratings." [Online]. Available: https://help.imdb.com/article/imdb/track-movies-tv/weighted-average-ratings/GWT2DSBYVT2F25SK?ref_=helpart_nav_8#.

[18] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[19] C. C. Aggarwal, *Recommender Systems The Textbook*. Switzerland: Springer International Publishing, 2016.

[20] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018.

[21] J. Han, M. Kamber, and J. Pei, *Data mining concepts and techniques*, Third Edit. Waltham: Elsevier Inc, 2012.

[22] P. Singh, *Deploy Machine Learning Models to Production*. Apress, 2021.

[23] K. Relan, *Building REST APIs with Flask*. Apress, 2019.

DOI: https://doi.org/10.29207/resti.v5i5.3486