Chatbot In The Selection Of Outpatient Ward In Hospital Using C4.5 Algorithm

Fandi Adi Prasetio
Department of Informatics
Universitas Jenderal Achmad Yani
West Java, Cimahi, Indonesia
fandiadi30@gmail.com

Faiza Renaldi. S.T., M.Sc, Fajri Rakhmat Umbara. S.T., M.T
Department of Information Systems
Universitas Jenderal Achmad Yani
West Java, Cimahi, Indonesia
faiza.renaldi@unjani.ac.id, fajri.umbara@gmail.com

Abstract

The C4.5 algorithm is one of many data mining classification techniques. The final result in the classification of this algorithm is the creation of a decision tree. *T*his tree helps in making predictions and selection. The c4.5 algorithm is widely used in many cases, such as outpatient services, disease prediction, *etc*. It's just that for the c4.5 algorithm method in the construction of chatbots, only a few apply it. More cases of implementing chatbots using Semantic Web, NLP, *etc*. This study focuses on the c4.5 algorithm method and its application in chatbots because, in previous studies, it was still rare to implement chatbot services in hospitals using this algorithm. The c4.5 algorithm method in this study will be carried out on a chatbot because in predicting poly hospital services for selection, it is more suitable to use this algorithm. The data used were obtained from the Azra Hospital Bogor, totaling 3,292 data after the data cleaning process with the classification results using the c4.5 algorithm, namely getting the RMSE (Root Mean Square) / Mean value of 12.66%. This number means that the lower the average prediction error, the less. That means that from the total data used, 315 data experienced wrong predictions, and 2,977 other data were predicted correctly. Gradient Boosting was also carried out to reduce the prediction error rate by 0.66% after being tested seven times. In the future, this research is expected to improve the classification test time.

Keyword
c4.5 algorithm, classification, chatbot, decision tree, outpatient clinic

1. Introduction

Chatbots have been widely used in the development of chat service technology. The advantages are that they can run without human supervision, minimize chat errors by admins, and provide quick responses at any time (Amato et al. 2017)(Oh et al. 2017). Chatbots assist users in doing many things, such as getting information, entertainment services, health services, customer services, and others (Mostaço et al. 2018). The chat response displayed to the user can vary depending on the concept and complexity of the technique itself. The more complex the program code, the more complex and diverse the responses that can be generated (Adamopoulou and Moussiades 2020).

The C4.5 algorithm method is used in this study to create a decision tree to implement the outpatient poly service chatbot (Mardi 2017). The c4.5 algorithm has been widely used in various fields, one of which is in health services such as: predicting emotions, healthcare system monitoring, and others (Nuzulia 2020)(Tamin and Iswari 2017). There are several advantages to using this algorithm. Namely, it can handle numeric attributes and missing values and can trim branches (Sharma and Kumar 2016). This algorithm was chosen because one study comparing the use of ID3, C4.5, and C5.0 found that the most powerful and preferred method in machine learning is C4.5 (Bimo et al. 2020).

The selected case study was outpatient poly service at the hospital in this study. Because there are still many limitations of the chatbot in its service to customer services (Heshmat and Eltawil 2016), this study focuses on predicting outpatient poly services and implementing them into a chatbot to help someone choose an online outpatient clinic service through a chatbot.

1.1 Objectives

The goal to be achieved in this study is to create a recommendation service model using the c4.5 algorithm, where this model will be used to predict from the entered keywords so that it can provide a response in the form of recommendation poly services such as poly services in outpatient dept.

2. Methods

This research method is divided into several parts, as shown in Figure 1

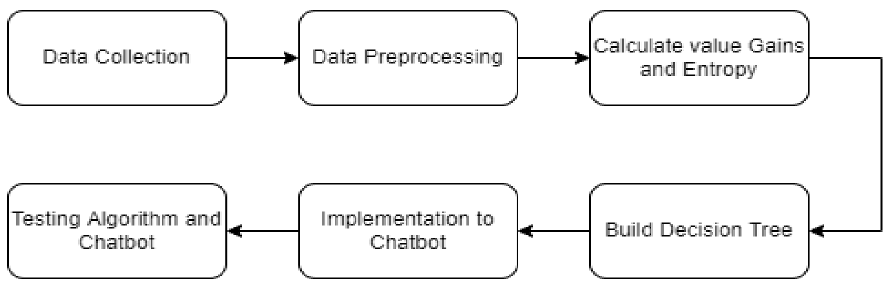

Figure 1. Research Methods

Divided into six steps.

Step 1 : Data Collection. Get outpatient service data from the hospital.

Step 2 : Data preprocessing. Pre-processing is required to be clean and prepare data for further analysis. Missing patient information should be completed or removed from the data

Step 3 : Calculate gains and Entropy. The data will calculate the gains and entropy values to be able to proceed to the next process.

Step 4 : Build a Decision Tree. After calculating the gains and entropy values, a decision will be made from the previous calculations.

Step 5 : Implementation Chatbot. The decision tree that has been made is then implemented into the chatbot.

Step 6 : Testing Algorithm and Chatbot. This step is to test whether the algorithm and chatbot can run well and what is the percentage error in service predictions.

2.1 Data Collection

The data was obtained from the Azra Bogor Hospital, Indonesia. The data used is outpatient service data. The data is a total of 5472 data with excel format files. The data obtained are such as patient name, registration number, date of admission, hospital code, address, gender, age, payment method, complaints, diagnosis code, and diagnosis information.

2.2 Data Pre-processing

To enter the calculation process of the c4.5 algorithm, the data must first go through preprocessing before it can be used. This process will be divided into three steps

Step 1 : Data Selection. This process will eliminate variables in the data that will not be used in the mining process. Examples of data that will not be used are registration number, patient entry date, hospital code, patient name, address, payment method, and diagnostic code. These variables will not affect the data mining process, so it should be removed. The data after the Selection process will be like Table 1.

Table 1. Result after Data Selection

no	jenis_poli	jk	usia	keluhan
1	Gigi & Mulut	Laki-laki	dewasa	dinding pipi sebelah dalam selalu tergigit gigi
2	THT	Perempuan	dewasa	radang tenggorokan
3	Syaraf	Perempuan	dewasa	sakit pingang
4	Syaraf	Perempuan	dewasa	sakit punggung
5	Penyakit Dalam	Perempuan	dewasa	sakit punggung dan sesak
6	THT	Perempuan	dewasa	tenggorokan sakit
7	Kesehatan Anak	Laki-laki	anak	3 hari demam tinggi
8	Kebidanan & Kandungan	Perempuan	dewasa	4d dan check up
9	Bedah Mulut	Perempuan	dewasa	abses
10	Bedah	Laki-laki		Abses Perianal

Step 2 : Data Cleaning. This process will divide the complaint variable into several parts because there are two, three or four complaints in one complaint variable. And another factor is to multiply the criteria needed in the mining process later. The complaint variable will be divided into two additional complaints so that it becomes complaint_1, complaint_2, and complaint_3. Changes are also made to the age attribute data, changing the year and month age data into categories of child and adult age (Expenditure 2021)(Indra et al. 2019).

Step 3 : Normalization. This step will only normalize the variables, which will later be used as a variable class for the c4.5 algorithm process to become numeric (Patro and Sahu 2015). the data that has reached this process will amount to 3292 data or objects. The results of the data that have been completed in preprocessing can be seen in Table 2.

Table 2. Result after Data Normalization

	jk	usia	keluhan_1	keluhan_2	keluhan_3	jenis_poli
0	Laki-laki	dewasa	TERGIGIT	GIGI	TIDAK ADA	8
1	Perempuan	dewasa	RADANG TENGGOROKAN	TIDAK ADA	TIDAK ADA	24
2	Perempuan	dewasa	SAKIT PINGGANG	TIDAK ADA	TIDAK ADA	23
3	Perempuan	dewasa	SAKIT PUNGGUNG	TIDAK ADA	TIDAK ADA	23
4	Perempuan	dewasa	SESAK	PUNGGUNG	TIDAK ADA	17
...	...	...	...	...	...	...
3287	Perempuan	anak	WAJAH ADA BRUNTUSAN	TIDAK ADA	TIDAK ADA	14
3288	Perempuan	anak	WAJAH BERUNTUSAN DAN PUTIH	TIDAK ADA	TIDAK ADA	14
3289	Perempuan	dewasa	WASIR	TIDAK ADA	TIDAK ADA	0
3290	Perempuan	dewasa	WASIR	TIDAK ADA	TIDAK ADA	0
3291	Perempuan	dewasa	WASIR	TIDAK ADA	TIDAK ADA	0

For further explanation regarding the intent or meaning of the attribute name, which is described in Table 3.

Table 3. Attribute Name Explained

No	Attribute	Type	Description
1	jk	String	This attribute describes gender and contains two instances: Laki-laki as Male and Perempuan as Female.
2	usia	String	This attribute describes age category and contains two instances: anak as child and dewasa as adult.

3	keluhan_1	String	This attribute describes complaints such as disease problems, this attribute has as many as 920 instances.
4	keluhan_2	String	This attribute is the same as the complaint attribute one because the patient usually does not only have one complaint.
5	keluhan_3	String	This attribute is the same as the complaint attribute one and two because the patient usually does not only have one complaint.
6	jenis_poli	String	This attribute describes the outpatient poly name and functions as a label attribute. This attribute has 26 instances, which means 26 types of outpatient poly.

2.3 Calculate the value of Entropy and Gains

In this stage, the entropy and gain values of each attribute and object will be calculated. This process aims to get value to build a decision at the next stage.

Step 1 : Calculate Entropy. First, count the total number of attributes, calculate the total number of attributes based on class *n* or their respective labels, and calculate entropy. This process is carried out continuously until all attributes have been calculated based on class n labels (Putri and Waspada 2018)(Fiandra, Defit, and Yuhandri 2017).

$$Entropy(X) = \sum_{j=1}^{k} p_j * \log_2 \frac{1}{p_j} = -\sum_{j=1}^{k} - p_j * log_2\, p_j \tag{1}$$

Step 2 : Calculate Gains. Select a variable or attribute as root based on the highest value of the existing variable. To calculate the gain value, use the formula in the equation below (Putri and Waspada 2018)(Fiandra, Defit, and Yuhandri 2017).

$$Gain(a) = Entropy\ (X) - \sum_{j=1}^{k} \frac{|X_i|}{X} * Entropy(X_i) \tag{2}$$

The gains value that has been obtained from the calculation of each variable or attribute can be seen in Table 4.

Table 4. Gains Value

Gains	
jk	0.00070438
usia	0.27681528
keluhan_1	3.81635313
keluhan_2	0.48680334
keluhan_3	0.09556318

2.4 Build Decision Tree

The result of this stage is making a decision tree. The decision tree that has been built will later be shaped like a Figure 2.

```
def findDecision(obj): #obj[0]: jk, obj[1]: usia, obj[2]: keluhan_1, obj[3]: keluhan_2, obj[4]: keluhan_3
    # {"feature": "keluhan_1", "instances": 3292, "metric_value": 3.8164, "depth": 1}
    if obj[2] == 'BATUK':
        # {"feature": "usia", "instances": 194, "metric_value": 0.9381, "depth": 2}
        if obj[1] == 'anak':
            # {"feature": "keluhan_2", "instances": 162, "metric_value": 0.1471, "depth": 3}
            if obj[3] == 'TIDAK ADA':
                # {"feature": "jk", "instances": 83, "metric_value": 0.3332, "depth": 4}
                if obj[0] == 'Laki-laki':
                    # {"feature": "keluhan_3", "instances": 49, "metric_value": 0.0, "depth": 5}
                    if obj[4] == 'TIDAK ADA':
                        return 11
                    else: return 11.26530612244898
                elif obj[0] == 'Perempuan':
                    return 11
                else: return 11.0
```

Figure 2. Result of Decision Built

The decision tree made is in the form of if-else if-else. In this function is explained every decision of each variable and its instance. Each variable or attribute is defined as a list object, with variable jk being object 0, usia being object 1, keluhan_1 being object 2, keluhan_2 being object 3, and keluhan_3 being object 4. This function also automatically creates entropy values, counts the number of data instances from variables, and measures how deep the tree is.

2.5 Implementation

This stage will explain how the algorithm and chatbot are implemented into the system to produce a function. At this stage, it will be divided into two parts: the algorithm's implementation into a prediction function and its implementation into the outpatient poly service selection chatbot.

2.5.1 Implementation Prediction

The implementation of the algorithm will be carried out using the decision tree function that has been built previously. In this implementation, an input form will be built for the input criteria, which will later be used as input for the recommendations for poly services that are displayed. The input used is jk, usia, keluhan_1, keluhan _2, keluhan _3. The input form will look like Figure 3.

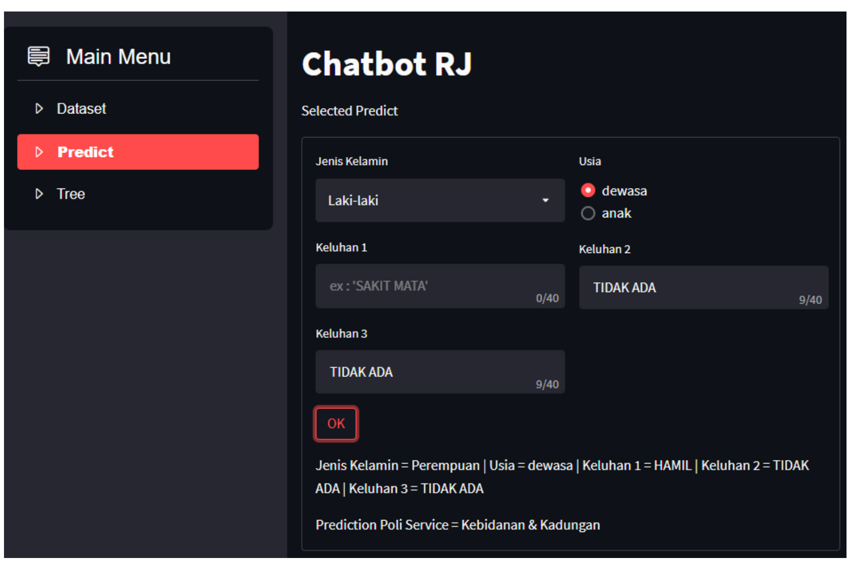

Figure 3. Implementation Predict

As in Figure 6, for example, the user will fill in the input. The final result of this process will display the name of the outpatient polyclinic that matches the criteria that have been entered: Kebidanan & Kandungan as Obstetrics & Gynecology.

2.5.2 Implementation Chatbot

Implementing the chatbot is done by connecting the decision tree that has been built and applying it to the chatbot. Here the chatbot development uses Telegram, so to start, you have to create code for the initial display and menus for selecting outpatient poly services. To see the chatbot interface can be seen in Figure 4.

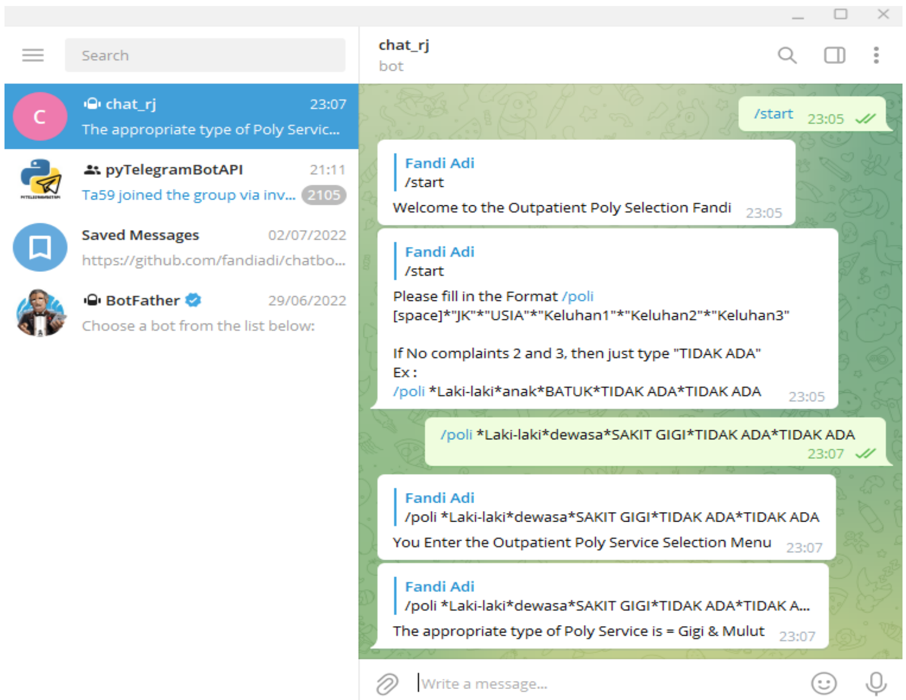

Figure 4. Implementation to Chatbot

For example, as in Figure 7, the user will start with the /start command, and then a chat will appear on how to use the service selection menu. The format used in this chatbot is to start with the /poli command. Then enter the criteria jk, usia, keluhan_1, keluhan _2, keluhan _3, each criterion separated by a star symbol. If so, a chat reply to the recommended poly menu will appear Gigi & Mulut as Teeth & Mouth.

3. Results and Discussion

Testing is carried out in two stages: on the algorithm to test the performance obtained and on the chatbot to test whether the chatbot can run well.

4.1 Testing Model Tree and Algorithm

In this stage, it will test whether the decision tree follows the results of calculations carried out by visualizing the decision as a tree. The root form is taken from the largest gain, complaint 1, then continued with the second-largest gain value, following the if-else decision function. If the results of the decision above are visualized into a decision tree image, it will be shaped like in Figure 5.

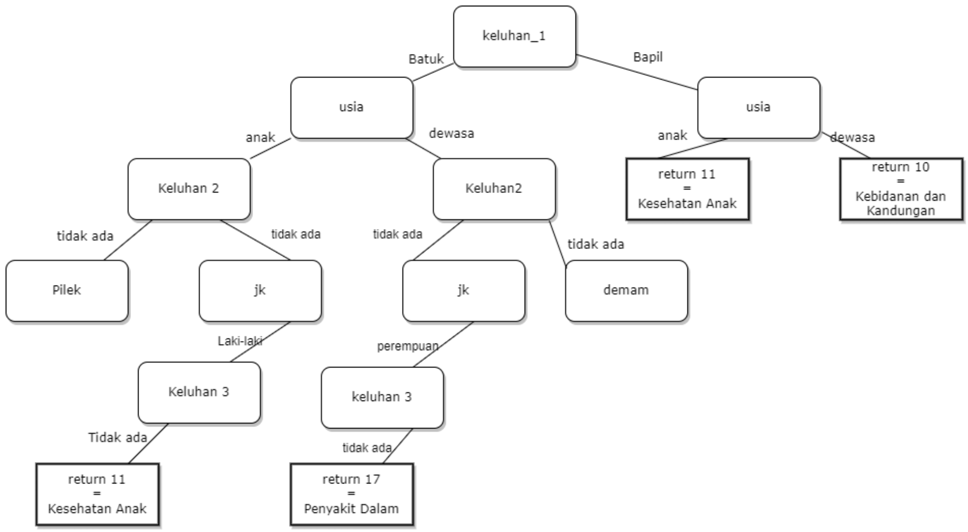

Figure 5. Visualized Decision Tree

Next is testing the performance of the algorithm. The test results obtained using this algorithm produce an RMSE value of 12.6653% with a mining process time of three minutes and thirty-five seconds. This means that the smaller the RMSE value, the better because the error in the prediction is reduced (Wang and Lu 2018). Results can be seen in Figure 6.

```
finished in  201.968811750412  seconds
-------------------------
Evaluate  train set
-------------------------
MAE:  0.38704941271769944
MSE:  2.459852166869178
RMSE:  1.5683915859469466
RAE:  0.11752664050853183
RRSE:  0.31531428817691226
Mean:  12.383353584447145
MAE / Mean:  3.125562151466091 %
RMSE / Mean:  12.665321839124143 %
```

Figure 6. Test Algorithm

A gradient boost is also performed to see if the prediction error value can be reduced by using this method (Wen et al. 2018)(Kadiyala and Kumar 2018). This test was repeated seven times, and the tree depth was five. The time required for this process is approximately twenty-two minutes. The results obtained are the RMSE value of 12,0005%. Gradient boost results can be seen in Figure 7.

```
finished in  1343.4021122455597  seconds
-------------------------
Evaluate  train set
-------------------------
MAE:  0.42123056427640937
MSE:  2.2084048693575937
RMSE:  1.4860702773952494
RAE:  0.11135793434928157
RRSE:  0.29876415806887663
Mean:  12.383353584447145
MAE / Mean:  3.40158715007099 %
RMSE / Mean:  12.000547890853067 %

{'trees': [<module 'outputs/rules/rules0' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules0.py'>,
  <module 'outputs/rules/rules1' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules1.py'>,
  <module 'outputs/rules/rules2' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules2.py'>,
  <module 'outputs/rules/rules3' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules3.py'>,
  <module 'outputs/rules/rules4' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules4.py'>,
  <module 'outputs/rules/rules5' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules5.py'>,
  <module 'outputs/rules/rules6' from 'C:\\Users\\fandi\\Python39\\chefboost\\Chatbot\\outputs/rules/rules6.py'>],
 'alphas': [],
 'config': {'enableGBM': True,
  'epochs': 7,
```

Figure 7. Test Algorithm with Gradient Boost

As in algorithm testing, the first test results are obtained to test the accuracy of the average error value and use a gradient boost to reduce the error value. There is less than a 1% difference between the first and boost tests. This is not a significant improvement, but it is a pretty good result because the value of the average prediction error rate can be reduced. That means that from the total data used, 315 data experienced wrong predictions, and 2,977 other data were predicted correctly. The graph of the test results can be seen in Figure 8.

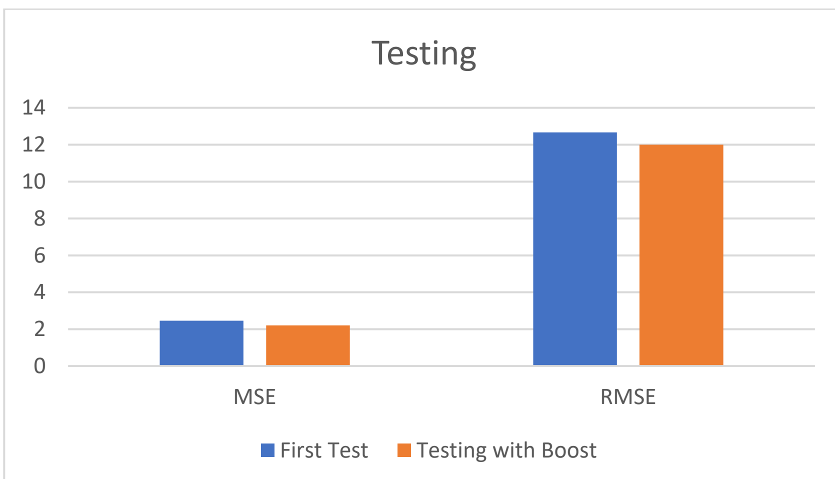

Figure 8. Testing Algorithm

4.1.2 Testing Chatbot

In testing of this chatbot, it will be carried out using the system functionality test method according to the test table below. Test testing is based on functionality that has been implemented into the chatbot (Okanović et al. 2020). Test questions are based on chat commands. Chatbots can function according to requirements and can provide appropriate responses. Chatbot testing can be seen in Table 5.

Table 5. Testing Chatbot

No	Test Questions	Result	Description/Reason
1	Can the chatbot react to the user?	Yes	When typing start or poli, it will respond according to the available menu.

2	Command /start chat can work properly?	Yes	This command will display a welcome message and a format message for using the /poli command.
3	The /poli command can provide a poly service response?	Yes	By following the applicable format, the chatbot can respond to the form of the outpatient poly name.
4	Can chatbots respond to anything other than /start and /poli ?	No	The chatbot now only supports the /start and /poli commands. Otherwise, the chatbot will not respond to anything.
5	The /poli chat command works fine	No	If, while filling out the input form and the user misses or does not fill in one of the input criteria, the chatbot will stop functioning.
6	Can chatbots handle complex questions?	No	Because the chatbot that was built is only intended for the development of the selection of poli services.

5. Conclusion

The research focused on implementing the c4.5 algorithm on the outpatient poly service chatbot. The applied algorithm focuses on processing outpatient medical record data in hospitals to provide options for poly service recommendations to chatbot users. Users can enter input criteria as a reference to the algorithm to choose the appropriate service for the user. The results of this study chatbot can provide a response in the form of an outpatient poly name to the user. It can be concluded that the application of the c4.5 algorithm in the selection of outpatient poly services in hospitals can be applied to the chatbot. With this application, it is hoped that users can be helped to choose the appropriate service through a chatbot.

References

Adamopoulou, Eleni, and Lefteris Moussiades. 584 IFIP IFIP Advances in Information and Communication Technology *An Overview of Chatbot Technology*. Springer International Publishing. 2020. http://dx.doi.org/10.1007/978-3-030-49186-4_31.

Amato, Flora et al. "Chatbots Meet Ehealth: Automatizing Healthcare." *CEUR Workshop Proceedings* 1982: 40–49.

Bimo, Panji et al. 2020. "Klasifikasi Dengan Pohon Keputusan Berbasis Algoritme C4.5." *PRISMA, Prosiding Seminar Nasional Matematika* 3: 64–71. 2017.

Expenditure, Health. "OECD Health Statistics 2021 Definitions , Sources and Methods." (July): 2021.

Fiandra, Yudha Aditya, Sarjon Defit, and Yuhandri Yuhandri. "Penerapan Algoritma C4.5 Untuk Klasifikasi Data Rekam Medis Berdasarkan International Classification Diseases (ICD-10)." *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 1(2): 82–89. 2017.

Heshmat, M, and A Eltawil. "Comparison between Outpatient Appointment Scheduling and Chemotherapy Outpatient Appointment Scheduling." *Egyptian Journal for Engineering Sciences and Technology* 19(1): 326–32. 2016.

Indra, Evta et al. "Application of C4.5 Algorithm for Cattle Disease Classification." *Journal of Physics: Conference Series* 1230(1). 2019.

Kadiyala, Akhil, and Ashok Kumar. "Applications of Python to Evaluate the Performance of Decision Tree-Based Boosting Algorithms." *Environmental Progress and Sustainable Energy* 37(2): 618–23. 2018.

Mardi, Yuli. "Data Mining : Klasifikasi Menggunakan Algoritma C4.5." *Edik Informatika* 2(2): 213–19. 2017.

Mostaço, Gustavo Marques et al. "AgronomoBot: A Smart Answering Chatbot Applied to Agricultural Sensor Networks Environmental Control and Automation for Swine Housing View Project." *14th International conference Precision Agri ulture* (June): 1–13. 2018. https://www.researchgate.net/publication/327212062.

Nuzulia, Maulida. "Pembuatan Fitur Chatbot Untuk Mengelola Emosi Dengan Menggunakan Algoritma C4.5 Berbasis Android." 2020.

Oh, Kyo Joong, Dongkun Lee, Byungsoo Ko, and Ho Jin Choi. "A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation." *Proceedings - 18th IEEE International Conference on Mobile Data Management, MDM 2017*: 371–76. 2017.

Okanović, Dušan et al. "Can a Chatbot Support Software Engineers with Load Testing? Approach and Experiences." *ICPE 2020 - Proceedings of the ACM/SPEC International Conference on Performance Engineering*: 120–29. 2020.

Patro, S.Gopal Krishna, and Kishore Kumar sahu. "Normalization: A Preprocessing Stage." *Iarjset*: 20–22. 2015.

Putri, Ratna Puspita Sari, and Indra Waspada. "Penerapan Algoritma C4.5 Pada Aplikasi Prediksi Kelulusan Mahasiswa Prodi Informatika." *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika* 4(1): 1. 2018.

Sharma, Himani, and Sunil Kumar. "A Survey on Decision Tree Algorithms of Classification in Data Mining." *International Journal of Science and Research (IJSR)* 5(4): 2094–97. 2016.

Tamin, Felix, and Ni Made Satvika Iswari. "Implementation of C4.5 Algorithm to Determine Hospital Readmission Rate of Diabetes Patient." *Proceedings of 2017 4th International Conference on New Media Studies, CONMEDIA 2017* 2018-Janua: 15–18. 2017.

Wang, Weijie, and Yanmin Lu. "Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model." *IOP Conference Series: Materials Science and Engineering* 324(1). 2018.

Wen, Zeyi et al. "Efficient Gradient Boosted Decision Tree Training on GPUs." *Proceedings - 2018 IEEE 32nd International Parallel and Distributed Processing Symposium, IPDPS 2018*: 234–43. 2018.

Biography

Faiza Renaldi is a lecturer in the Department of Information Systems, Faculty of Science and Informatics, Universitas Jenderal Achmad Yani Indonesia. He received his master of business informatics at Universiteit Utrecht, The Netherlands in 2006. His research interests are health informatics, information systems/information technology management, e-government, agile project management, and IT entrepreneurship.

Fajri Rakhmat Umbara Born in Palembang, earned a Diploma 3, Bachelor of Informatics at IT Telkom and continued his Masters in Informatics from Telkom University. Currently, he is a lecturer in the department of Informatics, Jenderal Achmad Yani University. His areas of interest are Software Engineering and Data Mining.

Fandi Adi Prasetio is a bachelor's degree student at the Universitas Jenderal Achmad Yani, West Java, Indonesia, and joined in informatics in 2017. His research interests are in the fields of web services and data mining.