Journal of Soft Computing Exploration

Comparation analysis of naïve bayes and decision tree C4.5 for caesarean section prediction

I Gusti Ayu Suciningsih[1], Muhammad Arif Hidayat [2], Renita Arianti Hapsari[3]
[12,3]Department of Computer Science, Universitas Negeri Semarang, Indonesia

Article Info

Article history:

Received Jan 8, 2021
Revised Feb 18, 2021
Accepted March 12, 2021

Keywords:

Caesar section
Pregnancy
C4.5
Naïve Bayes

ABSTRACT

The development of technology can be used to facilitate many matters. One of them is childbirth in the medical fields. Maternal mortality rate (MMR) is the number of maternal deaths during pregnancy to postpartum caused by pregnancy, childbirth or its management. There are several methods of labors that can be done. The determination of the labor is based on many factors and must be in accordance with the conditions of pregnant patient. Caesarean birth is the last alternative in labor, due to high risk factors. The objective of this research is to predicte and analyse caesarean section using C4.5 and Naïve Bayes classifier models. For experimentation the dataset is collected from UCI Machine Learning Repository and the main attributes represented in this dataset are age, delivery number, delivery time, blood of pressure, and heart problem. The accuracy using C4.5 by 80 training cases is 45% And the accuracy using Naïve Bayes is 50%.

Corresponding Author:

I Gusti Ayu Suciningsih,
Department of Computer Science,
Universitas Negeri Semarang,
Sekaran, Gunungpati, Semarang, Indonesia.
Email: suciningsihayu@gmail.com

1. INTRODUCTION

Maternal mortality rate (MMR) in Indonesia is still high. MMR represents number of maternal deaths during lifetime pregnancy until the post-term childbirth caused by pregnancy, childbirth and the puerperium or the management and not caused by accident or fell on every 100,000 live births [1]. Caesarean section is the last alternative in action labor. This is due to high risk factors, both risk for mother and babies [2]. Despite the high risk, numbers of caesarean birth experienced increase significantly, particularly in Indonesia. World Health Organization (WHO) set the standard for caesar section delivery in a country about 5-15 percent per thousand births in the world. Based on WHO data, in 2004-2008 in three continents (Latin America, Africa, and Asia) the lowest Caesarean birth rate was in Angola (2.3%) and the highest in China (46.2%). Caesar births data in Indonesia has increased sharply, especially in big cities. Lowest rate in Southeast Sulawesi (5.5%) and the highest in DKI Jakarta (27.2%) [3].

Information technology will continue to develop and needed to meet the needs of fast and accurate information for life [4]. Technology has been used in various fields, for example in the health sector [5]. At this time the health sector has been supported by technology that is able to visualize and predict a patient's condition. From existing patient data, it can be used as material to classify a patient's condition using technology. One area that requires classification of a patient's condition is a childbirth [6]. Based on the explanation above, it is necessaary to have an algorithm that can support the work of medical personnel in determining the type of labor [1]. Classification is one of the methods contained in data mining [7]. Classification is necessary to find patterns in order to be able to produce correct predictions even in critical conditions [8]. To perform the classification process, there are several algorithm that can be used including

Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbor (KNN), Decission Tree, and Artificial Neural Network (ANN). These methods have their own level of accuracy for each object to be classified. The methods that will be used in this project are comparation of the Decision Tree C4.5 and Naïve Bayes methods to classify the caesarean section.

2. METHOD

2.1 Application of Naïve Bayes Algorithm

This classifier is based on the Naïve Bayes Theorem, which gives a way to estimate the posterior probability. Posterior probability of a class gives the estimation of an item belonging to that class based on the given attributes. Naïve bayes is the simplest calculation of the Bayes theorem, because it is able to reduce computational complexity to simple multiplication of probability [9]. Apart from that, the Naïve Bayes algorithm is also capable of handling data sets which has many attributes.

The application of Caesarean Section data set on Naïve Bayes algorithm process as follows:

Prepare caesarean section data set.
Classifying using Naïve Bayes algorithm.
Count the number of classes or labels in the data set.
Count the number of cases on each class.
Multiply all the class variables.
Compare the results of each classes.
The following is the equation of the Naïve Bayes:

$P(H|X) = (P(X \mid H)P(H))/(P(X))$

In wich:

X : data or tuple object (class C)
H: : hypothesis
P(H|X) : probability that hypothesis H is in condition
P(H) : prior probability that the H hypothesis is valid (true)
P(X) : prior probability of tuple X.

2.2 Application of Decision Tree C4.5 Algorithm

The C4.5 algorithm [10] is used in Data Mining as a Decision Tree Classifier [11] which can be employed to generate a decision, based on a certain sample of data (univariate or multivariate predictors). The following is the application of the research Decision Tree C4.5 algorithm [12].

1. Determine the root of the tree.
2. Calculate entropy for the classes
3. Calculate entropy after split each attribute
4. Calculate information gain for each split
5. Perform the split
6. Perform further splits
7. Complete the decision tree

For choosing attribute as a root, based on the highest gain value of the exsting attributes. To calculate gain, a formula is used as shown in the equation:

Gain(S,A) = Entropy(S) 〖- Σ〗 _(i=1)^n (|S_i |)/(|S|) x Entropy(S_i)

In which:

S : case set
A : attribute
n : number of partitions attribute A
|S_i| : number of cases of i partitions
|S| : number of cases in S

2.3 Dataset Attribute Information

The dataset attribute on information can be seen at Table 1.

Table 1. Dataset attribute information

Attributes	Type	Description
Age	Integer	Age in years
Delivery Number	Integer	Birth stage
Delivery Time	Integer (0,1,2)	0 = Timely, 1= Premature, or 2 = Latecomer
Blood of Pressure	Integer (0,1,2)	0 = low, 1 = normal, or 2 = high
Heart Problem	Integer (0,1)	0 = apt, 1 = inept
Caesarean	Integer (0,1)	Whether patient is allowed to caesarean delivery. 0 = No or 1 = Yes

3. RESULT AND DISCUSSION

3.1 Sampel Data

The data isconsidered in ARFF format. The following gives the name of relation, name of attributes and sample instances in the given data set.
@attribute 'Age' {22,26,28,27,32,36,33,23,20,29,25,37,24,18,30,40,31,19,21,35,17, 38}
@attribute 'Age' {22, 26, 28, 27, 32, 36, 33, 23, 20, 29, 25, 37, 24, 18, 30, 40, 31, 19, 21, 35, 17, 38}, has 22 distinct values with a maximum value 40 and minimum value 17.
@attribute 'Delivery number' {1, 2, 3, 4}, considered up to the first four deliveries.
@attribute 'Delivery time' {0, 1, 2}, premature and late deliveries are taken into consideration.
@attribute 'Blood of Pressure' {2, 1, 0}, various blood pressure moods are noted at the timeof delivery.
@attribute 'Heart Problem' {1, 0}, heart response is apt or inapt.
@attribute Cesarean {0, 1}, a class attribute whether cesarean section delivery or not.

3.2 Cleaning Data

Cleaning data is checked on the dataset, if there is a missing value in the dataset, treatment must be given to the data [13]. In the dataset used for this study, there are no missing values as shown in Figure 1 which show a dataset of caesarean section, because there are no missing value then we can go to the next step .

	id	Age	Dev_number	Dev_time	Blood_preassure	Heart_problem	Caesarian
0	1	22	1	0	2	0	0
1	2	26	2	0	1	0	1
2	3	26	2	1	1	0	0
3	4	28	1	0	2	0	0
4	5	22	2	0	1	0	1

Figure 1. The dataset that shown in google colab

3.3 Determining Independet Variables and Dependent Variables

The dependent variable used here is the caesarian variable, because we want to see whether the patient is classified as caesar labor or normal labor. The other varible that are age, delivery number, delivery time, blood pressure, and heart problem became an independent variable, can be seen at figure 2.

```
[ ] # Variabel independen
    x = caesarian.drop(["Caesarian","id"], axis = 1)
    x.head()
```

	Age	Dev_number	Dev_time	Blood_preassure	Heart_problem
0	22	1	0	2	0
1	26	2	0	1	0
2	26	2	1	1	0
3	28	1	0	2	0
4	22	2	0	1	0

```
[ ] # Variabel dependen
    y = caesarian["Caesarian"]
    y.head()

0    0
1    1
2    0
3    0
4    1
Name: Caesarian, dtype: int64
```

Figure 2. The table of independent and dependent variable

3.4 Normalization

Normalization is rescaling real numeric attributes into range 0 and 1. That in the dataset there is data with values other than 0 and 2, then the normalization stage will be carried out so the data becomes values in the range 0 and 1.

```
[ ] from sklearn import preprocessing
    x1 = preprocessing.normalize(x)
    print (x1)

    [[0.99487439 0.04522156 0.         0.09044313 0.        ]
     [0.99632216 0.07664017 0.         0.03832008 0.        ]
     [0.99559146 0.07658396 0.03829198 0.03829198 0.        ]
     [0.9968264  0.03560094 0.         0.07120189 0.        ]
     [0.99487439 0.09044313 0.         0.04522156 0.        ]
     [0.99852398 0.03840477 0.03840477 0.         0.        ]
     [0.99658819 0.07382135 0.         0.03691067 0.        ]
     [0.99515266 0.09329556 0.         0.03109852 0.        ]
     [0.9968264  0.07120189 0.         0.03560094 0.        ]
     [0.99794872 0.03696106 0.03696106 0.03696106 0.        ]
     [0.99922929 0.02775637 0.         0.02775637 0.        ]
     [0.99908299 0.03027524 0.03027524 0.         0.        ]
     [0.99717646 0.0433555  0.0433555  0.0433555  0.        ]
     [0.99627096 0.04981355 0.         0.04981355 0.04981355]
     [0.99645179 0.03436041 0.06872081 0.         0.03436041]
     [0.99602384 0.03984095 0.07968191 0.         0.        ]
     [0.99840383 0.03993615 0.         0.03993615 0.        ]
     [0.98893635 0.04944682 0.09889364 0.09889364 0.        ]
     [0.99600652 0.08075729 0.         0.0269191  0.0269191 ]
     [0.99483201 0.04145133 0.08290267 0.         0.04145133]
     [0.99778842 0.03837648 0.03837648 0.03837648 0.        ]
     [0.99771219 0.06046741 0.         0.         0.0302337 ]
     [0.99523429 0.03980937 0.03980937 0.07961874 0.        ]
     [0.99863107 0.03698634 0.         0.         0.03698634]
     [0.99258333 0.04962917 0.         0.09925833 0.04962917]
     [0.9969278  0.05538488 0.         0.05538488 0.        ]
     [0.98936948 0.05496497 0.05496497 0.10992994 0.05496497]
     [0.99889074 0.03329636 0.         0.03329636 0.        ]
     [0.99708312 0.03115885 0.         0.0623177  0.03115885]
     [0.99486235 0.07652787 0.03826394 0.03826394 0.03826394]
     [0.99920096 0.03996804 0.         0.         0.        ]
     [0.99906382 0.0249766  0.         0.0249766  0.0249766 ]
     [0.99563423 0.06222714 0.         0.06222714 0.03111357]
     [0.99591    0.07377111 0.         0.03688556 0.03688556]
     [0.99340894 0.07641607 0.07641607 0.03820804 0.        ]
```

Figure 3. The stage of normalization

3.5 Data Testing and Data Training

The classification using naïve bayes is contained in the sklearn package [6]. In this classification, testing data and training data are needed. Dividing the data set into Data Testing and Data training aims to adjust the data set into the Algorithm model. Divided by the ratio of Data Training 75% and Data Testing 25%. Training data with random state is 123. The random state value is independent, the random state shows how many times the data is randomized. However, this time using 123 so that the random results we get are the same.

3.6 Calculate the Probability Value and the Predicted Results

```
[ ] # Menentukan probabilitas hasil prediksi
    nbtrain.predict_proba(x_test)

    array([[0.28134695, 0.71865305],
           [0.73095067, 0.26904933],
           [0.15167084, 0.84832916],
           [0.46259235, 0.53740765],
           [0.39529401, 0.60470599],
           [0.70104384, 0.29895616],
           [0.49926902, 0.50073098],
           [0.62370872, 0.37629128],
           [0.66441541, 0.33558459],
           [0.80417774, 0.19582226],
           [0.83222515, 0.16777485],
           [0.68946998, 0.31053002],
           [0.65620501, 0.34379499],
           [0.07948917, 0.92051083],
           [0.66441541, 0.33558459],
           [0.4562007 , 0.5437993 ],
           [0.53810831, 0.46189169],
           [0.61264038, 0.38735962],
           [0.03816595, 0.96183405],
           [0.10885903, 0.89114097]])
```

Figure 4. The result of probability values

The results that seen at Figure 4. For the example, the first data is 0.71 is rounded to 1, the second data is 0.26 is rounded to 0, and so on.

3.7 Confusion of Matrix

In figure 2. We can know that there are 5 pregnant women who are predicted to have normal labor and in actual circumstances do deliver normal. Meanwhile, the number of pregnant women who are predicted to have normal labor but in actual fact give birth by caesarean section is also 6. Then, there were 5 pregnant women who were predicted to give birth by caesarean section and in actual fact they gave birth by caesarean section. Meanwhile, there were 4 pregnant women who were predicted to give birth by caesarean section but in actual circumstances gave birth normally, the result can be seen at Figure 5.

prediction	0	1
actual		
0	5	4
1	6	5

Figure 5. The result confusion of matrix

3.8 Memory Usage

In the decision tree models memory that been used is 111,57 MB, shown in Figure 6.

```
[ ]  import os, psutil
     process = psutil.Process(os.getpid())
     print(process.memory_info().rss)  # in bytes

     111575040
```

Figure 6. The result of usage memory by Decision Tree C4.5

And in the naïve bayes algorithm, memory that been used is 111,70 MB, shown in Figure 7.

```
[ ]  import os, psutil
     process = psutil.Process(os.getpid())
     print(process.memory_info().rss)  # in bytes

     111706112
```

Figure 7. The result of usage memory by Naïve Bayes

3.9 Measure the Program Execution Time

In measuring the program execution time using a decision tree algorithm, the results are displayed for 0.013 seconds, shown in Figure 8.

```
[ ]  import timeit
     code_to_test = """
     a = range(100000)
     b = []
     for i in a:
         b.append(i*2)
     """
     elapsed_time = timeit.timeit(code_to_test, number=100)/100
     print(elapsed_time)

     0.013129985089999536
```

Figure 8. The result of execution time by Decision Tree C4.5

And in the execution time using naïve bayes algorithm the results are displayed for 0.012 seconds which means by using naïve bayes algorithm the execution time is faster, shown in Figure 9.

```
[ ]  a = range(100000)
     b = []
     for i in a:
         b.append(i*2)
```

```
[ ]  import timeit
     code_to_test = """
     a = range(100000)
     b = []
     for i in a:
         b.append(i*2)
     """
     elapsed_time = timeit.timeit(code_to_test, number=100)/100
     print(elapsed_time)

     0.01271887993999826
```

Figure 9. The result of execution time by Naïve Bayes Algorithm

3.10 Level of Accuracy

Decision tree models are created using 2 steps: Induction and Pruning. Induction is where we actually build the tree i.e set all of the hierarchial decision boundaries based on our data. Because of the nature of training decision tree they can be prone to mjor overfitting. Pruning is the process of removing the unnecessary structure from a decision tree, effectively reducing the complexity to combat overfitting with the added bonus of making it even easier to interpet. By using this method, the result of accuracy are shown in figure 10.

```
[ ]  from sklearn.tree import DecisionTreeClassifier
     from sklearn.metrics import accuracy_score
     from sklearn import tree
     clf_gini = DecisionTreeClassifier(criterion = "gini", random_state= 123, max_depth=3,min_sampl
     clf_gini.fit(x_train,y_train)
     y_pred = clf_gini.predict(x_test)
     from sklearn import metrics
     metrics.accuracy_score(y_test, y_pred)*100

     45.0
```

Figure 10. The result of accuracy in implementation by Decision Tree C4.5

After getting the Naïve Bayes algorithm classification model, calculate the accuracy using a confusion matrix. Naïve Bayes classification algorithm will produce better results if using more training data. The results of the accuracy in the Naïve Bayes classification are shown in Figure 11.

	precision	recall	f1-score	support
0	0.45	0.56	0.50	9
1	0.56	0.45	0.50	11
accuracy			0.50	20
macro avg	0.51	0.51	0.50	20
weighted avg	0.51	0.50	0.50	20

Figure 11. The result of accuracy in implementation by Naïve Bayes

4. CONCLUSION

Using C4.5 and Naïve Bayes classifier models , the result of accuracy are 45% and after getting the Naïve Bayes algorithm classification model, calculate the accuracy using a confusion matrix. Naïve Bayes classification algorithm will produce better results if using more training data. The results of the accuracy in the Naïve Bayes classification are 50%. So the level of accuracy using the Naïve Bayes method is greater or more accurate than the decision tree method.

REFERENCES

[1] H. Manik, M. F. G. Siregar, R. Kintoko Rochadi, E. Sudaryati, I. Yustina, and R. S. Triyoga, "Maternal mortality classification for health promotive in Dairi using machine learning approach," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 851, no. 1, 2020, doi: 10.1088/1757-899X/851/1/012055.

[2] Anggorowati and N. Sudiharjani, "Mobilisasi Dini dan Penyembuhan Luka Operasi Pada Ibu Post Sectio Caesarea (SC) di Ruang Dahlia Rumah Sakit Umum Daerah Kota Salatiga," *Pros. Semin. Nas.*

dan Int. Univ. Muhammadiyah Semarang, pp. 30–35, 2010, [Online]. Available: https://jurnal.unimus.ac.id/index.php/psn12012010/article/viewFile/1281/1334.

[3] L. Andayasari *et al.*, "Proporsi seksio sesarea dan faktor yang berhubungan dengan seksio sesarea di Proporsi Seksio Sesarea dan Faktor yang Berhubungan dengan Seksio Sesarea di Jakarta THE PROPORTION OF CAESAREAN SECTION AND ASSOCIATED FACTORS IN HOSPITAL OF JAKARTA," pp. 6–16, 2014.

[4] N. R. Indraswari and Y. I. Kurniawan, "Aplikasi Prediksi Usia Kelahiran Dengan Metode Naive Bayes," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 1, pp. 129–138, 2018, doi: 10.24176/simet.v9i1.1827.

[5] P. Radha and B. Srinivasan, "Predicting Diabetes by cosequencing the various Data Mining Classification Techniques," vol. 1, no. 6, pp. 334–339, 2014.

[6] A. Kamat, V. Oswal, and M. Datar, "Implementation of Classification Algorithms to Predict Mode of Delivery," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 5, pp. 4531–4534, 2015.

[7] R. H. Saputra and B. Prasetyo, "Improve the Accuracy of C4 . 5 Algorithm Using Particle Swarm Optimization (PSO) Feature Selection and Bagging Technique in Breast Cancer Diagnosis," pp. 47–55, 2020.

[8] M. W. L. Moreira, J. J. P. C. Rodrigues, A. M. B. Oliveira, K. Saleem, and A. V. Neto, "An inference mechanism using Bayes-based classifiers in pregnancy care," *2016 IEEE 18th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2016*, no. Dm, pp. 0–4, 2016, doi: 10.1109/HealthCom.2016.7749475.

[9] I. E. Tiffani, "Optimization of Naïve Bayes Classifier By Implemented Unigram , Bigram , Trigram for Sentiment Analysis of Hotel Review," pp. 1–7, 2020.

[10] A. Wibowo, D. Manongga, and H. D. Purnomo, "The Utilization of Naive Bayes and C.45 in Predicting The Timeliness of Students' Graduation," *Sci. J. Informatics*, vol. 7, no. 1, pp. 99–112, 2020, doi: 10.15294/sji.v7i1.24241.

[11] A. De Ramón Fernández, D. Ruiz Fernández, and M. T. Prieto Sánchez, "A decision support system for predicting the treatment of ectopic pregnancies," *Int. J. Med. Inform.*, vol. 129, pp. 198–204, 2019, doi: 10.1016/j.ijmedinf.2019.06.002.

[12] E. Fitriani, "Perbandingan Algoritma C4.5 Dan Naïve Bayes Untuk Menentukan Kelayakan Penerima Bantuan Program Keluarga Harapan," *Sistemasi*, vol. 9, no. 1, p. 103, 2020, doi: 10.32520/stmsi.v9i1.596.

[13] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, vol. 26-June-20, pp. 2201–2206, 2016, doi: 10.1145/2882903.2912574.