Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 216 (2023) 21–30

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Conference on Computer Science and Computational Intelligence 2022

Diabetes prediction using supervised machine learning

Muhammad Exell Febrian[a]*, Fransiskus Xaverius Ferdinan[a], Gustian Paul Sendani[a], Kristien Margi Suryanigrum[a], Rezki Yunanda[a]

[a]*Computer Science Department,School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia*

Abstract

Diabetes is a disease that can lead to blindness, kidney failure, and heart attacks, as well as death. According to the International Diabetes Federation, there were 463 million diabetics in 2019. If predictions are correct, this number will rise by 578 million by 2030, reaching 700 million by 2045. According to an article published by the Ministry of Health of the Republic of Indonesia in 2020, the ten countries with the highest diabetes rates in 2019 include Indonesia. The ability of experts is required to determine the type of diabetes disease. Because of their delay in discovering what disease they have, many people who are examined have a disease that can be described as severe. Diabetes detection technology is required to prevent severe conditions. In today's medical world, doctors can use it to quickly and accurately interpret diseases. Because of that we can use machine learning to prevent the death by making an artificial inteligent model that can predict diabetes disease and the method that be used is comparison between the KNN and Naive Bayes algorithms to see which algorithm suit the best for diabetes prediction. The study concluded by comparing two k-Nearest Neighbor algorithms and the Naive Bayes algorithm to predict diabetes based on several health attributes in the dataset using supervised machine learning. According to the results of our experiments and evaluating alghorithm using Confusion Matrix, the Naive Bayes algorithm outperforms KNN.

Keywords: diabetes; machine learning; artificial intelligent;

1. Introduction

Machine Learning is a branch of science that enables computers to be intelligent like humans, automatically improving their understanding through experience [1]. This branch of science focuses on systems that can learn on

* Corresponding author.
E-mail address: muhammad.febrian003@binus.ac.id

1877-0509
10.1016/j.procs.2022.12.107

their own, such as making decisions without being repeatedly programmed by humans. Not only that, but the machine can adapt to a constantly changing environment [1]. Machine learning is divided into four categories: Supervised Learning, Semi-Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Supervised Learning is a machine learning technique that is used for machine learning with labeled datasets in order to identify input labels in order to make predictions and classifications [1].

1.1. Research Problem

In this study, the research problem is because there were 463 million diabetics in 2019. If predictions are correct, this number will rise by 578 million by 2030, reaching 700 million by 2045. According to an article published by the Ministry of Health of the Republic of Indonesia in 2020, the ten countries with the highest diabetes rates in 2019 include Indonesia. The ability of experts is required to determine the type of diabetes disease. Because of their delay in discovering what disease they have, many people who are examined have a disease that can be described as severe. Because of this we will compare the KNN and Naive Bayes algorithms to see which algorithm suit the best for diabetes prediction. The object of this study is the Pima Indians Diabetes dataset, which contains 8 independent variables and 1 dependent variable. In this study, eight factors that are characteristic of diabetes are used to determine whether a person has diabetes or not, namely pregnancy, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age. With so many variables to consider in the diagnosis of diabetes, a method that produces results quickly and efficiently is required.

- To compare the classification accuracy of various machine learning algorithms if similar data is purchased.
- To learn how some machine learning algorithms classify.
- To categorize diabetes using a variety of machine learning algorithms
- Significance of the Study
- Contribute ideas for researchers and those interested in using machine learning to classify other diseases.
- To make a scientific contribution to the field of health sciences by raising readers' awareness of the characteristics of diabetes.

This article can be used as a resource for machine learning algorithms.

2. Literature review

2.1. Research Methods

This study use quantitative approaching which emphasizes the measurement of existing data. In conducting research, the stages of research are carried out as shown in Figure 1 as a reference [2]. The data used uses a public dataset originating from Kaggle with the name Pima Indians Diabetes Database dataset https://www.kaggle.com/uciml/pimaindians-diabetes-database (Learning, n.d.)

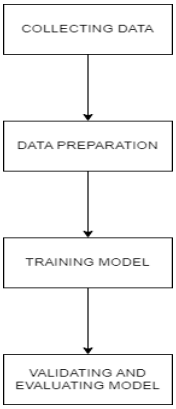

Fig 1. Research Framework

The Pima Indians Diabetes Database relates to the results of diagnosing diabetes carried out on female patients aged at least 21 years with 8 predictor attributes and outcome targets (0 or 1) as shown in Table 1. The Pima Indians diabetes database dataset consists of 768 data divided into 8 attributes. and 2 classes with a total of class 1 (268) and class 0 (500).

Table 1. Dataset that is used in this study

No	Attribute Name	Description
1	Pregnancies	Number of times pregnant
2	Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3	BloodPressure	Diastolic blood pressure (mm Hg)
4	SkinThickness	Triceps skin fold thickness (mm)
5	Insulin	2-Hour serum insulin (mu U/ml)
6	BMI	Body mass index (weight in kg/(height in m)^2)
7	DiabetesPedigreeFunction	Diabetes pedigree function
8	Age	Age (Years)
9	Outcome	Class Variable (0 or 1)

When the data is abtained, the preprocessing process is used by changing the type of the data from num to nom. Data that is already been converted will enter into the model experiment process. The result will be evaluated by comparing the test result with the result of the confusion matrix by looking at the accuracy, precision, and recall values. The values will be used as the model evaluation by taking the best value [3].

2.2. Proposed model classification

In this problem, the authors use two algorithm for comparing, K-Nearest Neighbour and Naïve Bayes. K-nearest Neighbor is one of the simplest Machine Learning Algorithm based on Supervised Learning techniques. The K-NN algorithm assumes the similarity between new cases/data with available cases and puts new cases into the category that is most similar to the available categories [4]. The K-NN algorithm stores all available data and classifies new data points based on similarity. This means that when new data appears it can be easily classified into the well suite category using the K-NN algorithm [5]. Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. This technique is easiest to understand when it is explained using binary or categorical input values. Called naive Bayes or idiot Bayes because the calculation of the probability for each hypothesis is simplified to make the calculation workable. This is a very strong assumption that is most unlikely in real data, namely that the attributes do not interact [6].

A. K-Nearest Neighbor

The KNN algorithm is one of the supervised learning techniques, it will calculate the proximity between the old

case (training data) and new cases (testing data) [7]. The steps of the KNN algorithm are:

- Determine the parameter k, as nearest neighbors (number of nearest neighbors);
- Calculate the square of the distance between the new data and the training data using Eucliden;
- Sort the closest neighbors that have the minimum distance;
- Classify the nearest neighbor according to the value of k;
- Determine the classification results based on the majority label.

B. Naïve Bayes

The Naive Bayes classifier is a simple probabilistic classifier based on the application of the Bayesian theorem (from Bayesian statistics) with the assumption of strong (naive) self-determination [8]. Naïve Bayes can also be called Simple Bayes and Independence Bayes. This algorithm can predict the probability of class membership, such as the probability of data given a certain class label. The Naive Bayes classifier assumes that the presence (or absence) of certain features (attributes) of a class is not related to the presence (or absence) of other features when a class variable is given [9]. The following is the Naïve Bayes equation:

$$p(C|F_1 \dots \dots F_n) = \frac{p(C)p(F_1 \dots F_n|C)}{p(F_1 \dots F_n)} \quad (1)$$

The advantages of using the Naïve Bayes algorithm are:

- Fast and highly scalable model
- Balancing linear with number of predictors and rows
- The Naïve Bayes procedure is parallel
- Naïve Bayes can be used for binary and multiclass classification.

2.3. Model testing and evaluation

For classification modeling, each experiment was carried out using a split factor data set (training data: test data) from 80 to 10. From each experiment, 6 different models and 2 identical models were obtained [10]. The parameters for testing each modeling output are as follows: accuracy, precision, and recall. The final result will be a comparison of which algorithm is better between the two algorithms, namely the KNN algorithm or Naive Bayes.

A. K-Fold Cross Validation

In k-fold cross validation, the dataset is divided randomly by K parts. To generate each section, one of the K sections is used as validation and combines the remaining from the K-1 section into training data [11]. The following is the equation of k-fold cross validation:

$$\begin{aligned} v_1 &= x_1 \mathcal{T}_1 = x_2 \cup x_3 \cup \dots \cup x_k \\ v_2 &= x_2 \mathcal{T}_2 = x_1 \cup x_3 \cup \dots \cup x_k \\ &\dots \\ v_k &= x_k \mathcal{T}_k = x_1 \cup x_2 \cup \dots \cup x_{k-1} \end{aligned} \quad (2)$$

The use of 10 fold cross validation because this method is a standard method in practice [12]. The dataset is divided into 10 parts, where this dataset consists of 2 types, namely training data and testing data. Figure 2 is an illustration of 10 fold cross validation.

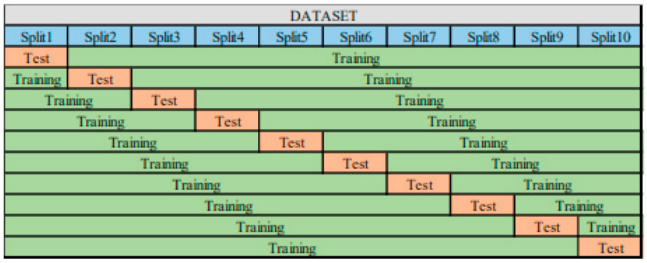

Fig 2. 10 Fold cross validation dataset

3. Prepare the model

3.1. Data Collection

The data in this study uses a public dataset originating from Kaggle that can be access in (https://www.kaggle.com/uciml/pimaindians-diabetes-database) with the name of the dataset used using a public dataset originating from Kaggle with the dataset name Pima Indians Diabetes Database. The Pima Indians diabetes database dataset consists of 768 data which is divided into 8 attributes and 2 classes with a total of class 1 (268) and class 0 (500).

3.2. Data Pre-Processing

Preprocessing is the initial process that will transform the input data into data with the appropriate format and ready to be processed. Some examples of things that are done in preprocessing include various processes that are needed, including: merging, changing shapes, or transforming data as a way to clean, integrate, reduce and discretize. Furthermore, the process in preprocessing can consist of one process activity or a combination of several processes above. The existing process depends on the goals to be achieved in the preprocessing. Selection of the right process needs to be done considering that the appropriate process in the data preprocessing stage will improve classification performance. Thus, to improve the quality of the data to be analyzed, data preprocessing steps should be carried out. To create quality data, the author has already done this preprocessing. This step involves processes such as Data Integration, Data Cleaning, Data Reduction, Data Transformation.

3.3. Data Cleaning

This step refers to finding incomplete, incorrect, inaccuratedata, and missing data (Missing Value). Although in this dataset none of the columns contain missing values, some of the measurements (Glucose, Blood Pressure, Skin Thickness, Insulin and BMI) have values of 0 (Table 2), which is not possible for a living human organism.

Table 2. Columns and count of 0

Column	Count of 0
Glucose	5
BloodPressure	35
Skin Thickness	227
Insulin	374
BMI	11

3.4. Data Integration

At this step, data integration is needed to change the measurement scale of the original data into another form

so that the analysis tool can read the diabetes dataset. However, at this step the author does not use this stage because the diabetes dataset is already in the form of useful data for the analysis step

3.5. Data Reduction

This stage requires a data reduction process to increase storage and reduce time and cost. However, at this step the author does not use this stage because the diabetes dataset is already in the form of useful data for the analysis step.

3.6. Data Transformation

This step will change the data into a form that is suitable with the analysis method. This step requires data transformation to convert the original measurement data into a form so that the analyser can read the diabetes dataset. However, at this step the author does not use this stage because the diabetes dataset is already in the form of useful data for the analysis step

4. Model testing and evaluation

4.1. Model Testing

For the test of KNN and Naïve Bayes models, we used Google Collaboratory to replicate Jupyter Notebook in the cloud. Here is the test model that we did at Google Collaboratory

- Importing model that we need to test the study concluded by comparing two k-Nearest Neighbor algorithms and the Naive Bayes algorithm to predict diabetes based on several health attributes in the dataset using
- Performing data pre-processing stages such as reading the dataset, viewing the contents of the dataset, and filling in the blank data in the dataset
- Performing data scaling or data normalization which aims to convert the numerical values in the dataset to a common scale, without distorting differences in the range of values. Normalization of data will help speed up the learning process in machine learning, as well as separate training data and testing data
- Testing the KNN model
- Evaluating the KNN model with a confusion matrix to see the values of accuracy, precision, and recall as well as doing cross validation so that the experiment can be validated.
- Testing the Naïve Bayes model
- Evaluating the Naïve Bayes model with a confusion matrix to see the values of accuracy, precision, and recall as well as doing cross validation so that the experiment can be validated.
- Visualizing of the comparison of the values of accuracy, precision, and recall from eight experiments with the percentage distribution of datasets

4.2. Model Evaluation

We have successfully used the KNN and Naive Bayes algorithms to predict diabetes based on several health attributes. The percentage of attribute data is compiled from 80% to 10% training data. Table 2 shows the classification accuracy for the KNN and Naive Bayes algorithms in eight experiments.

From the table below, it can be seen that from eight experiments, the best accuracy was in the first experiment when the training data was 80%. The KNN algorithm has produced a good accuracy of 77.92%, while Naive Bayes has produced a good accuracy of 78.57%. In eight trials, the best accuracy was in the first experiment with 80% training data. For the cross-validation section, we have added it as one to validate the results of the accuracy values we got. The next experiment has a detailed comparison between KNN and Naive Bayes in terms of accuracy for each training data allocation. Naive Bayes was more accurate in all experiments. Naive Bayes has a higher average than KNN with an accuracy of 76.07% than KNN 73.33%. Table 3. shows the comparative accuracy of KNN and Naive Bayes in the allocation of training data.

Table 3. Classification Accuracy from KKN and Naïve Bayes Model

Attempt	Training Percentage	KNN			Naïve Bayes		
		Accuracy	*Recall*	*Precision*	*Accuracy*	*Recall*	*Precision*
1	80%	77,92%	75%	74%	78,57%	74%	75%
2	70%	77,92%	73%	75%	76,19%	70%	73%
3	60%	74,03%	69%	71%	75,65%	69%	73%
4	50%	73,96%	70%	71%	75,78%	71%	73%
5	40%	73,54%	70%	71%	75,49%	71%	73%
6	30%	70,07%	67%	67%	76,58%	73%	75%
7	20%	68,13%	64%	65%	75,45%	71%	73%
8	10%	71,10%	67%	68%	74,86%	72%	72%
Average		73,33%	69,37%	70,25%	76,07%	71,37%	73,37%

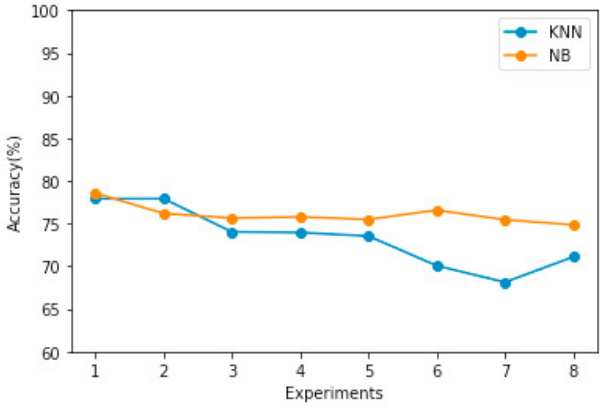

Fig 3. Accuracy between KNN and Naïve Bayes

Based on the Figure 3 above, we can see that the accuracy value between the KNN and Naïve Bayes models continues to change based on the percentage distribution of the dataset.

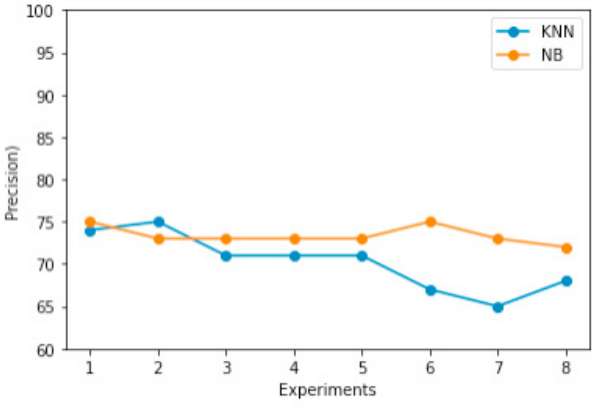

Fig 4. Precision between KNN and Naïve Bayes

Based on the Figure 4 above, we can see that the precision value between the KNN and Naïve Bayes models continues

to change because of the different confusion matrix in each experiment based on the percentage distribution of the dataset.

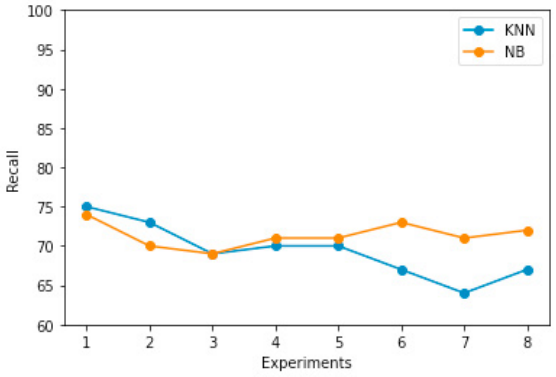

Fig 5. Recall between KNN and Naïve Bayes

Based on the Figure 5 above, it can be seen that the recall value between the KNN and Naïve Bayes models continues to change due to the different confusion matrix in each experiment based on the percentage distribution of the dataset. Here's a KNN method comparison between this research and a reference about Irish flower:

Table 4. KNN method comparison between this research and a reference about Irish flower

KNN Method Comparison		
	This study	**Irish flower research**
1	Determine the parameter	Calling library
2	Calculate the square of the distance between the new data and the training data	Calling dataset
3	Sort the closest neighbors that have the minimum distance	Activating algorithm and data fitting
4	Classify the nearest neighbor according to the value of k	Visualize the result
5	Determine the classification results based on the majority label	

And Figure 6 show the result comparison between this research and a reference about Irish flower:

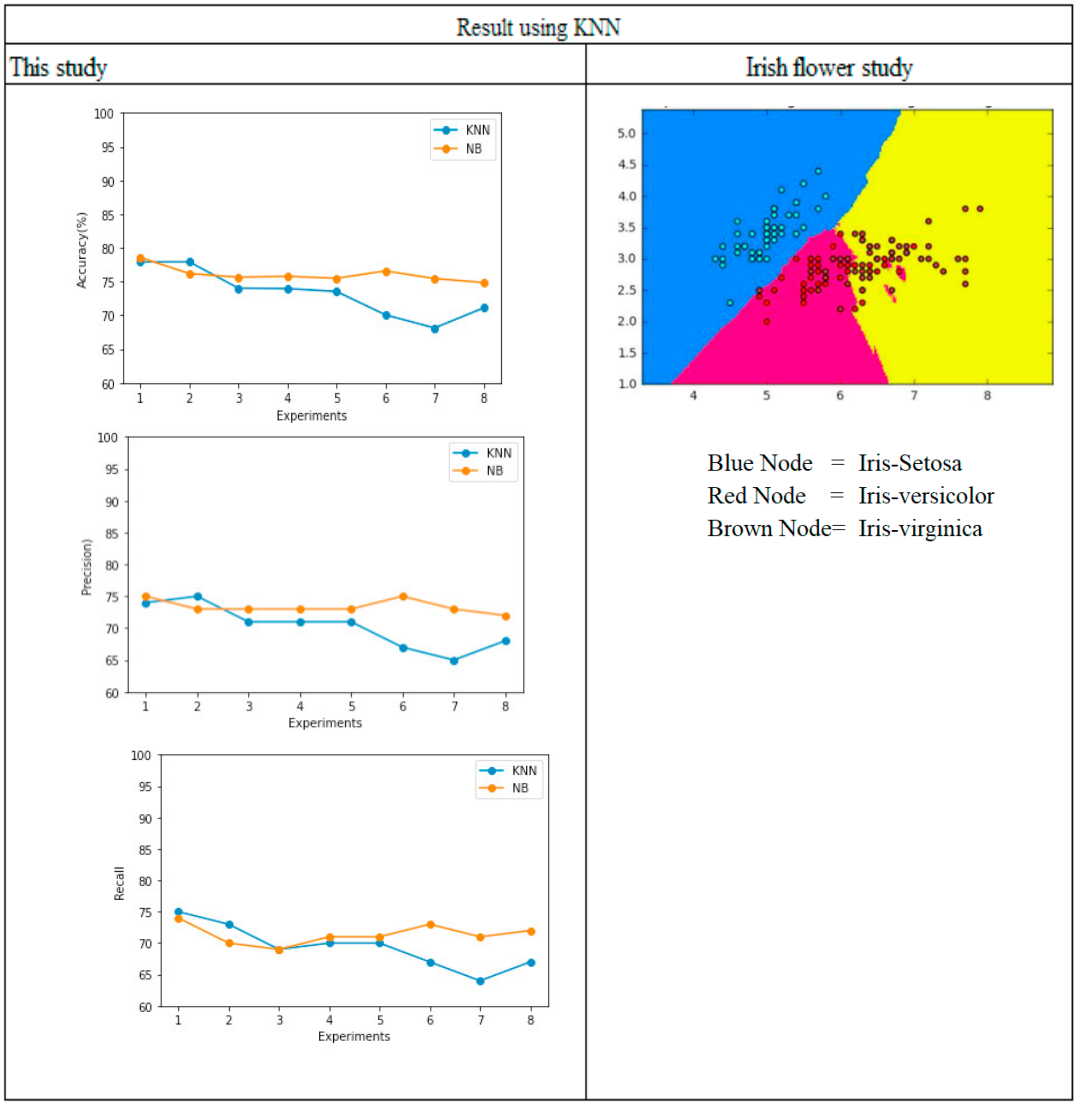

Fig 6. Result comparison between this research and a reference about Irish flower

5. Conclusion and suggestions

The study concluded by comparing two k-Nearest Neighbor algorithms and the Naive Bayes algorithm to predict diabetes based on several health attributes in the dataset using supervised machine learning. According to the results of our experiments and evaluating alghorithm using Confusion Matrix, the Naive Bayes algorithm outperforms KNN, with an average value of 76.07 percent accuracy, 73.37 percent precision, and 71.37 percent recall in Naive Bayes and an average value of 73.33 percent accuracy, precision 70.25 percent, and recall of 69.37 percent in KNN. As a result, it can be concluded that the Naive Bayes algorithm is preferable to the KNN algorithm for predicting diabetes using the Pima Indians dataset. For future research can be done by adding other algorithm like neural network and other techniques in order to produce an accuracy value and better precision also by Adding technique Particle Swarm Optimization for optimize the results and using application program development.

References

[1] R. P. Endang Retnoningsih, Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python, Bekasi: BINA INSANI ICT JOURNAL, 2020.

[2] Hadi, A. F., Setiawidayat, S., & Qustoniah, A. (2018). Perancangan Dan Pembuatan Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Diabetes Mellitus Berbasis Android. Jurnal WIDYA TEKNIKA, 26(1), 1–11.

[3] S. D. Jadhav and H. P. Channe, "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques," Int. J. Sci. Res., vol. 5, no. 1, pp. 1842–1845, 2016.

[4] R. S. Wahono, N. S. Herman, and S. Ahmad, "Neural Network Parameter Optimization Based on Genetic Algorithm for Software Defect Prediction," vol. 20, no. 10, pp. 1951–1955, 2014.

[5] Karthick, R. & Malathi, D.A., 2015. Preprocessing of Various Data Sets Using Different Classification Algorithms for Evolutionary Programming. International Journal of Science and Research (IJSR), IV(4), pp.2730-33.

[6] X. Wu, S. Wang, and Y. Zhang, "Review of K nearest neighbor algorithm theory and application," Computer Engineering and Application, vol. 53, no. 21, pp. 1–7, 2017.

[7] V. Sindhu, S. A. S. Prabha, S. Veni, and M. Hemalatha, "Thoracic surgery analysis using data mining techniques," vol. 5, no. April, pp. 578–586, 2014.

[8] W. Xu, L. Jiang, An attribute value frequency-based instance weighting filter for naive Bayes [J]. Journal of Experimental & Theoretical Artificial Intelligence 31(4), 225–236 (2019).

[9] S. Sugriyono and M. U. Siregar , "Prapemrosesan klasifikasi algoritme kNN menggunakan K-means dan matriks jarak untuk dataset hasil studi mahasiswa," Jurnal Teknologi dan Sistem Komputer, vol. 8, no. 4, pp.311-316, 2020

[10] Anand, R., Kirar, V.P.S. & Burse, K., 2013. K-Fold Cross Validation and Classification Accuracy of PIMA Indian Diabetes Data Set Using Higher Order Neural Network and PCA. International Journal of Soft Computing and Engineering (IJSCE, II(6), pp.436-38.

[11] Riadi, A. (2017). Penerapan Metode Certainty Factor Untuk Sistem Pakar Diagnosa Penyakit Diabetes Melitus Pada Rsud Bumi Panua Kabupaten Pohuwato. ILKOM Jurnal Ilmiah, 9(3), 309–316. https://doi.org/10.33096/ilkom.v9i3.162. 309-316.

[12] Alpaydın Ethem, Introduction to Machine Learning Second Edition, 2nd ed. London: MIT, 2010.