Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2024. Vol. 4. No. 1; e-ISSN: 2808-4519

Application of the K-Nearest Neighbor Method for Classification of Hypertension Diseases (Case Study: Stabat Health Center)

Indah Kelana Sari[1*], A M H Pardede[2], Magdalena Simanjuntak[3]

[1, 2, 3]STMIK Kaputama

indahkelanasari1@gmail.com[1], akimmhp@live.com[2], magdalena.simanjuntak84@gmail.com[3]*

Abstract

Globally, the WHO (World Health Organization) estimates that non-communicable diseases cause about 60% of deaths and 43% of diseases worldwide. Hypertension is a disease that occurs due to an increase in blood pressure in humans. It is difficult to know if a person has hypertension, without measuring the patient's blood pressure. According to the American Heart Association (AHA), the number of Americans over the age of 20 suffering from hypertension has reached 74.5 million, but nearly 90-95% of cases have no known cause. It is estimated that about 80% of the increase in hypertension cases will occur mainly in developing countries by 2025, from 639 million cases in 2000. This number is expected to increase to 1.15 billion cases in 2023. This study uses a quantitative approach with experimental methods to test the application of K-Nearest Neighbor (KNN) in the classification of hypertension diseases at the Stabat Health Center. The description of the results obtained is to make the right decision regarding when and how to treat the disease to prevent the worst possibility for patients by classifying the severity of hypertension both in normal circumstances, prehypertension, stage 1 hypertension, and stage 2 hypertension. The results of the trial show that the KNN model is able to provide accurate predictions based on patient history data available at Stabat Health Center.

Keywords: *Hypertension, K-Nearest Neighbor*

1. Introduction

This study aims to utilize the KNN method to group patient data based on various parameters (such as blood pressure, age, weight, and others). So that it can identify patients who suffer from hypertension or not. The application of KNN can help improve the accuracy of hypertension diagnosis by analyzing patient data more carefully, thereby reducing the possibility of misdiagnosis because in this day and age many hypertensive patients, not only the elderly but adolescents now suffer from hypertension. This is caused by unhealthy dietary factors (excessive salt consumption, a diet high in saturated fats and trans fats, low intake of fruits and vegetables), lack of physical activity, tobacco and alcohol consumption, and overweight or obesity.

Hypertension is high blood pressure or a condition that indicates systolic blood pressure >140 mmHg or diastolic blood pressure ≥90 mmHg. Data mining is a process that uses one or more machine learning techniques to analyze and extract knowledge automatically. Data mining uses a variety of data analysis software to find patterns and relationships in data so that it can be quickly used to make accurate predictions. Data mining is the activity of finding interesting patterns from large amounts of data, data can be stored in databases, data warehouses, or other information storage. Data mining is related to other fields of science, such as database systems, data warehousing, statistics, machine learning, information retrieval, and high-level computing. In addition, data mining is supported by other sciences such as neural networks, pattern recognition, spatial data analysis, image databases, signal processing.

K-Nearet Neighbor (K-NN) is a method for classifying objects based on the learning data closest to the object. K-Nearet Neighbor is based on the concept of learning by analogy. The learning data is described with n-dimensional numerical attributes. Each learning data represents a point, denoted by c, in an n-dimensional space. If a data query with an unknown label is entered, then K-Nearest Neighbor will look for the training data closest to the query data in n-dimensional space.

Python is a versatile interpretive programming language with a design philosophy that focuses on the readability level of the code. Python is claimed to be a language that combines capabilities, capabilities, with a very clear code syntax, and comes with the functionality of a large and comprehensive standard library.

2. System Analysis and Design

2.1. Research Methods

This study uses a quantitative approach with experimental methods to test the application of K-Nearest Neighbor (KNN) in the classification of hypertension diseases at the Stabat Health Center.

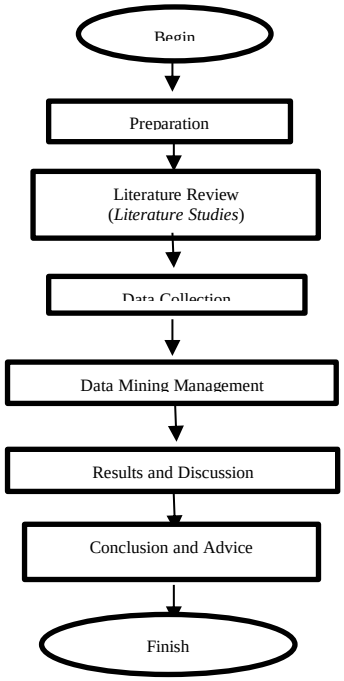

Fig. 1: Research workflow

2.2. Data Sources and Attributes

The initial stage carried out in this study is to prepare data, where data is obtained from the Stabat Health Center in 2024. The data used is data on hypertension patients in 2024. The data obtained will be used in this study in the form of data related to hypertension attributes, namely age, weight, systolic blood pressure, and diastolic blood pressure. The total data used was 519 records with normal classes, prehypertension, stage 1 hypertension, and stage 2 hypertension. Of the 519 data, it will be divided into 415 training data and 104 test data. The data will be calculated using the K-Nearest Neighbor method. From the data owned, it was selected as attributes of age, weight, systolic blood pressure, diastolic blood pressure.

2.3. Research supporting data

The data used to support the research decision were 15 data with attributes used for age, weight, systolic blood pressure, diastolic blood pressure as shown in table 1:

Stage 1.
processing research datasets with data from the Stabat Health Center.

Table 1: research dataset

Number	Age	Systolic	Diastolic	Weight	Class
1.	57	137	85	55	Prehypertension
2.	65	142	79	87	Stage 1
3.	64	213	111	79	Stage 2
4.	73	192	121	55	Stage 2
5.	61	214	122	59	Stage 2
6.	57	151	88	53	Stage 1
7.	63	162	98	52	Stage 2
8.	64	200	100	76	Stage 2
9.	73	150	70	52	Stage 1
10.	63	176	96	62	Stage 1
11.	61	154	79	52	Stage 1
12.	68	144	82	56	Stage 1
13.	49	244	128	55	Stage 2
14.	73	173	87	72.1	Stage 2
15.	71	180	90	52	Stage 2

Stage 2.
At this stage of normalization, the goal is to narrow or shrink the range values in the data. The following is the normalization formula:
Normalisasi xando = (x-min (xk))))/(max (xk) – min (xk) ..... 2.1)
Information:
Xk = normalized outcome value
X = value of x before normalization
min = minimum value of the attribute
max = maximum value of the attribute.

Table 2: Finding the Max and Min Values

Number	Age	Systolic	Diastolic	Weight
MAX	88	244	134	108
MIN	17	99	60	39

From the average and max values, normalization will be carried out from the first patient test data in Table 3.4 as follows:
Pasien_uji(Age) = (57-17)/(88-17) =40/71 = 0.56
Pasien_uji(systolic) = (137-99)/(244-99) = 38/145 = 0.41
Pasein_uji(Diastolic) = (85-60)/(134-60) = 25/74 = 0.34
Pasien_uji(Weight) = (55-39)/(108-39) = 16/69 = 0.23

Stage 3.
Calculation of the Euclidean distance
The following is an example of distance calculation using the following formula:
D (a ,b) = √(〖(X1-X2)〗 ^2+ 〖(Y1-Y2)〗 ^2) . . . . . (2.2)
Information:
D (a,b) = Jarak Euclidean Data a and Data b
X = X point coordinates (Test data of each attribute)
Y = Y point coordinates (Train Data for Each Attribute)
Example of test data:
D (Test Data, Training Data)
= (UsiaDU – UsiaDL)2 + (SistolikDU – SistolikDL)2 + (DiastolikDU – DiastolikDL)2 + (BBDU – BBDL)2
= (0,69 - 0,81)2 + (0,42 - 0,84)2 + (0,41 - 0,49)2 + (0,24 - 0,01)2
= 0,0144 + 0,1764 + 0,0064 + 0,0529
= √0,2501 = 0,5004
The following is table 3 of the results of the calculation of 15 data.

Table 3: Euclidean Distance Search Results

Number	Age	Systolic	Diastolic	Heavy	*Euclidean*	Class
1	0,111805556	1,233333333	0.0059	0,353472222	3,475	Stage 2
2	0,269444444	1,115277778	0,490277778	0,225694444	3,819444444	Stage 2
3	0,309722222	1,193055556	0,970138889	0.0013	4,151388889	Stage 2
4	0,1375	0.0018	0.0040	0,734722222	2,516666667	Stage 2
5	0.0008	0.0057	0.0010	0.0062	0,811111111	Stage 1
6	0,309722222	0,120833333	0,834722222	0.0087	3,034027778	Stage 2
7	0,1375	0,607638889	0,338888889	0.0014	2,755555556	Usual
8	0,309722222	0,402083333	0,088194444	0.0062	2,420138889	Prehypertension
9	0,352083333	0.0005	0,352083333	0.0005	2,222222222	Stage 2
10	1,238888889	0,176388889	0.0052	0,871527778	4,015972222	Stage 2
11	1,003472222	0.0017	0,490277778	0,304166667	3,544444444	Stage 1
12	0.0097	0.0049	0.0052	0.0000	0,977083333	Stage 2
13	0,088194444	0.0005	0.0098	0.0087	1,236805556	Stage 2
14	0,166666667	0,108333333	0,095138889	0.0065	1,697222222	Stage 2
15	0,088194444	0.0011	0.0040	1,63125	3,490277778	Stage 1

Based on the results obtained, the value of K is:

Table 4: value of K

K	Accuracy
1	97,12%
2	95,19%
3	95,19%
4	97,12%
5	96,15%

3. Discussion and Implementation

3.1. Component testing

The purpose of testing python components on a jupyter notebook involves verifying that the python code is running correctly and working as expected in a jupyter environment.

3.1.1. Creating a Dataset

Importing panda as pd is the first step to using the pandas library in a python script. This allows us to access various functions of the panda with the abbreviated name pd.

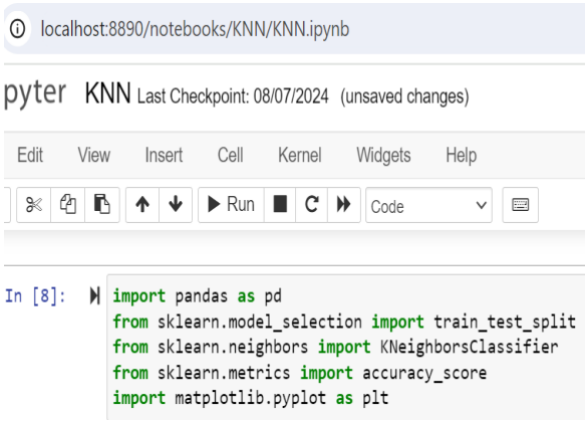

Fig. 2: Pandas Script

3.1.2. Loading a Dataset

Load the data into the pandas DataFrame and preprocess it if needed.

Fig. 3: Dataset Loading Script

3.1.3. Data Withdrawal

So we have to call the data so that we know whether the data we are importing is correct or not.

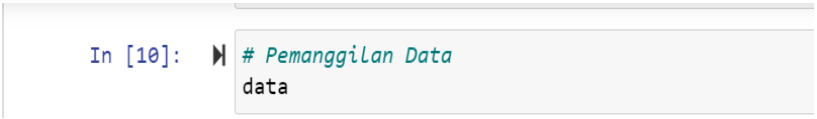

Fig. 4: Data Call

3.1.4. Separating Features and Labels

Works for all columns except the last one is the feature (y) and the label (x).

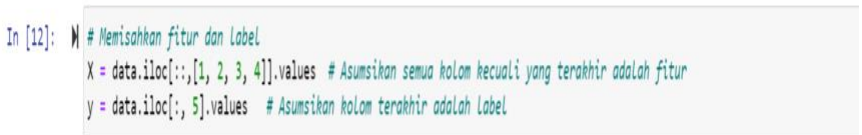

Fig. 5: Separating Features and Labels

3.1.5. Dividing the Data Set into Training Data and Test Data

So that the model can be evaluated properly.

Fig. 6: Dividing the Data Set into Training Data and Test Data

3.1.6. Training the KNN Model

By training the KNN model, it can determine the number of closest neighbors.

```
In [15]:  # Melatih model
          knn.fit(X_train, y_train)
```

Fig. 7: KNN Model Training

3.1.7. Predicting and Evaluating Models

We can use it to make predictions on test data and evaluate its performance.

```
In [16]:  # Memprediksi label untuk data uji
          y_pred = knn.predict(X_test)

In [24]:  # Daftar untuk menyimpan akurasi
          k_values = range(1, 15)
          accuracies = []
```

Fig. 8: Predicting Labels

3.1.8. Determining the Value of K

Choosing the optimal k value is an important step in KNN. One approach is to try to share the value of k and plot its accuracy.

```
In [25]:  # Menghitung akurasi untuk setiap nilai k
          for k in k_values:
              knn = KNeighborsClassifier(n_neighbors=k)
              knn.fit(X_train, y_train)
              y_pred = knn.predict(X_test)
              accuracy = accuracy_score(y_test, y_pred)
              accuracies.append(accuracy)
              print(f'k = {k}, Akurasi = {accuracy * 100:.2f}%')

k = 1, Akurasi = 97.12%
k = 2, Akurasi = 95.19%
k = 3, Akurasi = 95.19%
k = 4, Akurasi = 97.12%
k = 5, Akurasi = 96.15%
k = 6, Akurasi = 95.19%
k = 7, Akurasi = 94.23%
k = 8, Akurasi = 95.19%
k = 9, Akurasi = 95.19%
k = 10, Akurasi = 95.19%
k = 11, Akurasi = 93.27%
k = 12, Akurasi = 96.15%
k = 13, Akurasi = 96.15%
k = 14, Akurasi = 94.23%
```

Fig. 9: Determining the K Value

3.1.9. Graph Plot Accuracy Against K-Values

Create graphs to visualize the relationship between k-value and model accuracy.

```
In [27]:  # Membuat grafik plot akurasi terhadap nilai k
          plt.figure(figsize=(10, 6))
          plt.plot(k_values, accuracies, marker='o', linestyle='-', color='b')
          plt.xlabel('Nilai K')
          plt.ylabel('Akurasi')
          plt.title('Akurasi KNN untuk Berbagai Nilai K')
          plt.xticks(k_values)
          plt.grid(True)
          plt.show()
```

Fig. 10: Script for Creating Charts

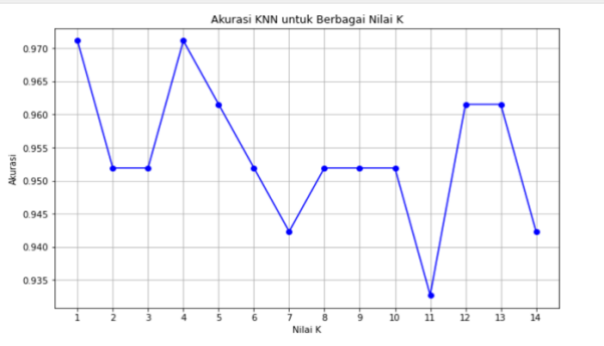

Fig. 11: Graph Results

3.1.10. Calculating Accuracy

Calculate accuracy by comparing predicted labels (y_pred) with actual labels (y_test).

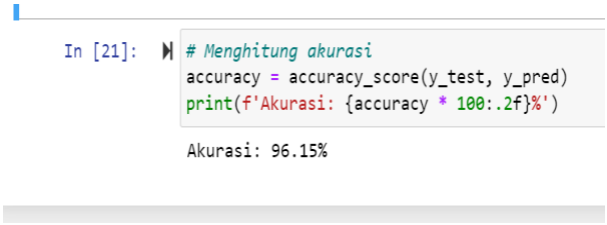

Fig. 12: Calculating Accuracy

4. Conclusion

From the results of the research that has been carried out regarding the application of the K-Nearest Neighbor method for the classification of hypertension disease (case study: Stabat Health Center), the following conclusions can be drawn:

1. Accuracy of the KNN Model: The KNN method has been shown to be effective in classifying hypertension diseases with a high degree of accuracy. The results of the trial show that the KNN model is able to provide accurate predictions based on patient history data available at Stabat Health Center.
2. K Optimization: Graphs help us find the optimal K value. For example, if we want the model and the highest accuracy, we can choose K=1, K=3, or K=7 depending on the trade-off between model complexity and generalization.
3. Accuracy Stability: Although K=1 has the highest accuracy, a higher K value such as K=7 can provide more stable and robust results on invisible data, as the model relies less on one closest neighbor.
4. Implementation in Puskesmas: The implementation of the KNN-based system at the Stabat Health Center shows great potential to assist medical personnel in the early detection and classification of hypertension disease. With this system, the diagnosis process can be carried out faster and more accurately, so that it can improve the quality of health services.

Reference

[1] Cholil, S. R., Handayani, T., Prathivi, R., & Ardianita, T. (2021). Application of the K-Nearest Neighbor (KNN) classification algorithm for Scholarship Recipient Selection Classification. IJCIT (Indonesia Journal of Computers and Information Technology), 6(2), 118–127. https://doi.org/10.31294/ijcit.v6i2.10438

[2] A. M.H. Pardede, M. Zarlis, and H. Mawengkang, "Optimization of Health Care Services with Limited Resources," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 4, pp. 1444–1449, 2019, doi: 10.18517/ijaseit.9.4.8348.

[3] Dr. Vladimir, VF (2015). Relations in the Field of Data Mining Science, Man. Gastronomía ecuatoriana y turismo local., 1(69), 5–24.S.P. Mohanty, U. Choppali, and E. Kougianos, "Everything you want to know about smart cities," *IEEE Consum. Electron. Stomach.*, vol. 5, no. 3, pp. 60–70, 2016, doi: 10.1109/MCE.2016.2556879.

[4] Gusriani, Haniarti, & Henni Kumaladewi Hengky. (2021). The Effect of the Risk of Hypertension Incidence on Police Members at the Parepare Police. Scientific Journal of Human and Health, 4(1), 101–109. https://doi.org/10.31850/makes.v4i1.402D. Niyigena, C. Habineza, and TS Ustun, "Computer-based smart energy management system for rural health centers," 2016, doi: 10.1109/IRSEC.2015.7455005.

[5] Hartati, Y. (2018). The Effect of Brisk Walking Exercises on Lowering Blood Pressure in Hypertensive Patients at Andalas Health Center, Padang City. Nursing, 2(02), 1–10.

[6] Jasmir, Abidin, D. Z., Nurmaini, S., & Malik, R. F. (2017). Application of the K-Nearest Neighbor Method in Predicting the Student Study Period (Case Study: STIKOM Dinamika Bangsa Students). Proceedings of the Annual Research Seminar, 3(1), 133–138.

[7] Makawekes, E., Suling, L., & Kallo, V. (2020). Effect of physical activity on blood pressure in the elderly 60-74 years. Journal of Nursing, 8(1), 83. https://doi.org/10.35790/jkp.v8i1.28415

[8] https://www.kajianpustaka.com/2017/09/data-mining.html

[9] https://sardjito.co.id/2022/08/31/ayo-kendalikan-hipertensi/#:~:text=Normal%20apabila%20sistolik%20%3C%20120%20mmHg,dan%20diastolic%20%2F%20%C2%B3%20100%20mmHg

[10] Mustika and Sudiantara. (2022). Source: (Mustika and Sudiantara, 2019). 8–21.

[11] Richard Oliver (in Zeithml., et al. 2018). (2021). Hypertension Literature Review. Angewandte Chemie International Edition, 6(11), 951–952., 2013–2015.

[12] Widiyanto, A., Atmojo, J. T., Fajriah, A. S., Putri, S. I., & Akbar, P. S. (2020). Hypertension Prevention Health Education. Jurnalempathy.Com, 1(2), 172–181. https://doi.org/10.37341/jurnalempathy.v1i2.27

[13] Yuli Mardi. (2019). Data Mining: Classification Using C4 Algorithm. 5 Data mining is part of the stages of the Knowledge Discovery in Database (KDD) process. Journal of Informatics Education. Journal of Informatics Education, 2(2), 213–219