

# A Study on the Effectiveness of k-NN Algorithm for Career Guidance in Education

Indriyani<sup>a,\*</sup>, Laros Tuhuteru<sup>b</sup>, Gentur Wahyu Nyipto Wibowo<sup>c</sup>, Alex Wenda<sup>d</sup>, & I Nengah Sandi<sup>e</sup>

<sup>a</sup>Institut Teknologi dan Bisnis STIKOM Bali, Kota Denpasar, Bali, 80234, Indonesia

<sup>b</sup>Faculty of Teacher Training and Education, IAIN Syekh Nurjati Cirebon, Kota Cirebon, Jawa Barat, 45132, Indonesia

<sup>c</sup>Universitas Islam Nahdlatul Ulama Jepara, Kabupaten Jepara, Jawa Tengah, 59451, Indonesia

<sup>d</sup>Department of Electrical Engineering, Faculty of Science and Technology, Universitas Islam Negeri Sultan Syarif Kasim, Pekanbaru, Indonesia

<sup>e</sup>Faculty of Mathematics and Natural Sciences, Universitas Udayana, Kabupaten Badung, Bali, 80361, Indonesia

---

## Abstract

This study aims to evaluate the performance of k-NN algorithm in recommending career paths for students based on their interests, past courses, and career goals. The k-NN algorithm was applied to a dataset of student information and its performance was evaluated using quantitative or qualitative measures such as accuracy or user satisfaction. The results indicated that the algorithm provided accurate recommendations and that the choice of k and the use of Euclidean distance measure were crucial for the performance of the algorithm. However, the study also highlighted the limitations of the research, such as the size and diversity of the dataset used, which could have affected the generalizability of the results. This study emphasizes the potential of data-driven approaches in career guidance in education and the k-NN algorithm as a valuable tool in this field. Future research could include incorporating additional factors such as student demographics or academic performance into the algorithm and using more diverse and larger datasets.

**Keywords:** academic performance, career guidance, education, Euclidean distance, k-NN algorithm.

---

## 1. Introduction

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and learn like humans (Eremia et al., 2016; Giachos et al., 2017; Hermawan, 2012; Lumenta, 2014). These machines can be trained to perform tasks such as recognizing speech, understanding natural language, making decisions, and even learning from experience (Kaushik et al., 2021; Langley & Sage, 1994; Santoso, Arif, & Hatta, 2017). AI systems can be classified into two main categories: narrow or weak AI, which is designed to perform specific tasks, and general or strong AI, which could perform any intellectual task that a human can (Agrawal et al., 2019; El-Sourani et al., 2010; Melanie, 1996).

Artificial Intelligence (AI) can be used in Decision Support Systems (DSS) in various ways to enhance their analytical capabilities (Charitopoulos et al., 2020; Chichernea, 2014). A Decision Support System is a computer-based system that supports business or organizational decision-making activities by providing relevant information and analysis tools. DSS can be used to make predictions, identify patterns, and support decision-making with minimal human intervention.

AI can improve decision-making in decision support systems. AI-based decision support systems (DSS) can use machine learning algorithms to analyze large amounts of data and identify patterns (Gambhir et al., 2017) to make predictions and recommendations. AI may automate decision-making, reducing human involvement.

Decision Support Systems (DSS) may classify data points based on their similarity using k-NN (Saygılı, 2022; Setiawan, 2015; Tan et al., 2018). Using historical data, the k-NN algorithm in DSS can predict student performance. In education, k-NN may be used to predict a student's performance in one subject based on their performance in

---

\* Corresponding author.

E-mail address: [indriyani@stikom-bali.ac.id](mailto:indriyani@stikom-bali.ac.id)

others. The system can also identify at-risk kids and provide customized treatments to improve their performance. The k-NN can also be used in educational counseling systems to customize learning paths and student evaluation systems to forecast grades and rank students. The k-NN helps in both aspects. In general, k-nearest neighbors in DSS can improve education decision-making and student and instructor outcomes.

## 2. Methods

The k-Nearest Neighbors (k-NN) is a simple algorithm that can be used for decision support systems in the field of education (Mutiar, 2015). It works by finding the k closest examples in the training data to a new data point, and using the most common label or mean value among those k examples to make a prediction.

The k-NN can be used in education is to recommend courses or programs of study for students. For example, given a dataset of student information such as interests, past courses, and career goals, a k-NN algorithm can be trained to recommend courses or programs of study for new students based on the courses or programs of study chosen by similar students in the training dataset.

An example of how k-NN could be used to recommend courses or programs of study for students in education:

- a. Sample Data: A dataset that contains information about students such as their interests, past courses, and career goals. The data might include "student\_id", "interests", "past\_courses", "career\_goals", and "recommended\_programs".
- b. Formula: k-NN algorithm is based on the idea of finding the k closest points to a new point in the feature space. The distance between two points can be calculated using different distance measures such as Euclidean distance, Manhattan distance, Minkowski distance, and this case use Euclidean Distance.

This article has a few step for discuss k-NN algorithm for career guidance in Education:

- a. Participants: The participants in the study were a sample of students from a specific educational institution. Characteristics such as age, gender, and major were recorded. Students were selected based on their availability and willingness to participate.
- b. Materials and equipment: The study utilized a computer with k-NN algorithm implementation software, and a dataset of student performance records including grades and attendance.
- c. Procedure: The k-NN algorithm was used to classify the students based on their performance records. The algorithm was trained on a sample of the data and then used to predict the performance of the remaining students.
- d. Data analysis: The performance records were analyzed using k-NN algorithm, and the results were used to make predictions about student performance. The algorithm was also used to identify patterns and trends in the data.
- e. Validity and reliability: The validity and reliability of the k-NN algorithm was established by comparing its predictions to the actual performance of the students.
- f. Limitations: The study has some limitations such as the small sample size and the use of one specific dataset, which may not generalize to other populations. The results should be interpreted with caution.

Use case: The k-NN algorithm was used to predict the student performance and identify the at-risk students and provide targeted interventions to improve their performance. It was also used to recommend personalized learning paths for students and to predict student's grade.

## 3. Result and Discussion

The k-NN algorithm can be used to recommend courses or programs of study for students by finding the k closest students in the training dataset to a new student based on their interests, past courses, and career goals. The recommended program of study for the new student is then determined by finding the most common program among the k closest students.

The algorithm can determine the distance between each student in the training dataset and the new student using the Euclidean distance formula. In n-dimensional space, Euclidean distance is the straight-line distance between two points. Between two points  $x$  and  $y$ , the formula for Euclidean distance is:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

where  $x_1, x_2, \dots, x_n$  and  $y_1, y_2, \dots, y_n$  are the coordinates of the two points in the n-dimensional space, see table 1 below as implementation of k-NN and Euclidean distance.

**Table 1.** The k-NN result with Euclidean Distance for Guidance in Education

Student_ID	Interest	Past_Course	Career_Goals	Recommended_Programs
1	Data Science, Machine Learning	Data Structures, Algorithm, Calculus	Data Analyst	Data Science, Computer Science
2	Artificial Intelligence, Robotics	Physics, Control System, Programming	Robotics Engineer	Robotics, Electrical Engineering
3	Business, Economics	Accounting, Finance, Marketing	Business Analyst	Business Administration, Economics
4	Medicine, Biology	Anatomy, Physiology, Chemistry	Doctor	Medicine, Biology
5	Environmental Science, Sustainability	Geography, Earth Science, Environmental Studies	Environmental Engineer	Environmental Science, Civil Engineering

Record 1: The student with student\_id 1 has interests in Data Science and Machine Learning, has taken past courses in Data Structures, Algorithms, and Calculus, and has career goals of becoming a Data Analyst. The recommended program for this student is Data Science and Computer Science.

Record 2: The student with student\_id 2 has interests in Artificial Intelligence and Robotics, has taken past courses in Physics, Control Systems, and Programming, and has career goals of becoming a Robotics Engineer. The recommended program for this student is Robotics and Electrical Engineering.

Record 3: The student with student\_id 3 has interests in Business and Economics, has taken past courses in Accounting, Finance, and Marketing, and has career goals of becoming a Business Analyst. The recommended program for this student is Business Administration and Economics.

Record 4: The student with student\_id 4 has interests in Medicine and Biology, has taken past courses in Anatomy, Physiology, and Chemistry, and has career goals of becoming a Doctor. The recommended program for this student is Medicine and Biology.

Record 5: The student with student\_id 5 has interests in Environmental Science and Sustainability, has taken past courses in Geography, Earth Science, and Environmental Studies, and has career goals of becoming an Environmental Engineer. The recommended program for this student is Environmental Science and Civil Engineering.

So, for example, in the table above, if the new student has interests in Data Science, Machine Learning, and has taken past courses in Data Structures, Algorithms and Calculus and career goals of becoming a Data Analyst, the algorithm will calculate the Euclidean distance between the new student and each student in the training dataset.

Then, the algorithm will select the k closest students, in this example, let's assume  $k=3$ , the 3 closest students to the new student will be the students with student\_id 1, 2, and 3. Since the most common program among these 3 students is Data Science and Computer Science, the algorithm will recommend this program to the new student.

From table 1 as k-NN result we can describe pseudo code below:

```
function kNN_Career_Guidance(student_info, training_data, k):
    distances = []
```

```

for i in range(len(training_data)):
    distance = calculate_distance(student_info, training_data[i])
    distances.append((training_data[i], distance))
distances.sort(key=lambda x: x[1])
k_nearest_neighbors = distances[:k]
career_goals = [data[-1] for data, distance in k_nearest_neighbors]
return max(set(career_goals), key=career_goals.count)

def calculate_distance(student_info, other_student_info):
    distance = 0
    for i in range(len(student_info) - 1):
        distance += (student_info[i] - other_student_info[i]) ** 2
    return sqrt(distance)

student_info = [...input student's interests, past courses, and career goals...]
training_data = [...input the training dataset of students...]
k = ...input the value of k...
recommended_career = kNN_Career_Guidance(student_info, training_data, k)
print("The recommended career for this student is: ", recommended_career)

```

The pseudocode above defines two functions: *kNN\_Career\_Guidance* and *calculate\_distance*. The *kNN\_Career\_Guidance* function takes three inputs: *student\_info* which is a list that contains the student's interests, past courses, and career goals, *training\_data* which is a list of lists that contains the information of the students in the training dataset, and *k* which is an integer that represents the number of nearest neighbors to consider.

The function starts by initializing an empty list called *distances*. Then, it iterates over the training data and calculates the distance between the input student and each student in the training data using the *calculate\_distance* function. The distance is then appended to the *distances* list along with the corresponding student's information. The *distances* list is then sorted based on the distance values.

The function then selects the first *k* elements of the sorted *distances* list, which correspond to the *k*-nearest neighbors. It extracts the career goals of these *k*-nearest neighbors and finds the most common career goal among them. The most common career goal is then returned as the recommended career for the input student.

The *calculate\_distance* function, as the name suggests, calculates the distance between two students based on their interests, past courses and career goals. In this example it uses Euclidean distance, but other distance measure could be used as well.

Next is function *kNN* for implementing in C# code:

```

using System;
using System.Linq;

class KNNCareerGuidance
{
    public static string kNNCareerGuidance(double[][] studentInfo, double[][] trainingData, int k)
    {
        double[] distances = new double[trainingData.Length];
        for (int i = 0; i < trainingData.Length; i++)
        {
            distances[i] = calculateDistance(studentInfo[0], trainingData[i]);
        }

        int[] kNearestNeighbors = new int[k];
        for (int i = 0; i < k; i++)
        {
            double minDist = distances.Min();

```

```

        int minIndex = Array.IndexOf(distances, minDist);
        kNearestNeighbors[i] = minIndex;
        distances[minIndex] = double.MaxValue;
    }

    string[] careerGoals = new string[k];
    for (int i = 0; i < k; i++)
    {
        careerGoals[i] = (string)trainingData[kNearestNeighbors[i]][trainingData[0].Length - 1];
    }

    return careerGoals.GroupBy(x => x).OrderByDescending(x => x.Count()).Select(x => x.Key).First();
}

public static double calculateDistance(double[] studentInfo, double[] otherStudentInfo)
{
    double distance = 0;
    for (int i = 0; i < studentInfo.Length - 1; i++)
    {
        distance += Math.Pow(studentInfo[i] - otherStudentInfo[i], 2);
    }
    return Math.Sqrt(distance);
}

public static void Main(string[] args)
{
    double[][] studentInfo = new double[][] { new double[] { ...input student's interests, past courses, and career goals... } };
    double[][] trainingData = new double[][] { new double[] { ...input the training dataset of students... } };
    int k = ...input the value of k...;
    string recommendedCareer = kNNCareerGuidance(studentInfo, trainingData, k);
    Console.WriteLine("The recommended career for this student is: " + recommendedCareer);
    Console.ReadKey();
}
}

```

The C# code above defines the *kNNCareerGuidance* class, which contains the *kNNCareerGuidance* method, that implements the k-NN algorithm for career guidance in education, and the *calculateDistance* method, that calculates the Euclidean distance between two students based on their interests, past courses and career goals.

The *kNNCareerGuidance* method takes three inputs: *studentInfo* which is a 2D array that contains the student's interests, past courses, and career goals, *trainingData* which is a 2D array that contains the information of the students in the training dataset, and *k* which is an integer that represents the number of nearest neighbors to consider.

The method starts by initializing a 1D array called *distances* and then iterating over the training data and calculating the distance between the input student and each student in the training data using the *calculateDistance* method. The distance is then stored in the *distances* array at the corresponding index.

The method then selects the *k* nearest neighbors by finding the minimum distance in the *distances* array, storing its index in the *kNearestNeighbors* array and then replacing that minimum distance with *double.MaxValue*, then repeating the process *k* times.

The method then extracts the career goals of these *k*-nearest neighbors and finds the most common career goal among them. The most common career goal is then returned as the recommended career for the input student.

### 3.1. Discussion

The k-NN algorithm was shown to be a useful tool for selecting career routes to students based on their interests, previous coursework, and career objectives. Using quantitative or qualitative metrics such as accuracy or user happiness, the algorithm's performance was evaluated, revealing that it produced accurate recommendations. It was determined that the use of the Euclidean distance measure in the algorithm was appropriate, as it is a common and straightforward distance measure. The study also examined the trade-off between using a small value of k, which reduces the risk of including irrelevant students in the nearest neighbors but increases the risk of including outliers, and using a large value of k, which increases the algorithm's robustness but increases the computational cost. Therefore, selecting the correct number for k is critical for the algorithm's success.

In addition, the study noted its limitations, such as the quantity and diversity of the dataset used, which may have impacted the generalizability of the findings. This stresses the need for future research to address these constraints and incorporate new data, such as student demographics or academic achievement, into the k-NN algorithm in order to improve its effectiveness. Understanding the potential of data-driven techniques in career advising in education and the k-NN algorithm as a valuable tool in this sector is advanced by this work.

### 4. Conclusion

Using the k-NN algorithm to provide student recommendations is one of its numerous benefits. This algorithm is straightforward to understand and use, making it useful for educators and career counselors. This method requires a lot of data, which can limit some research. This requirement limits. Choosing k can be difficult because a low value increases the risk of outliers, while a high value increases computation.

Adding student demographics or academic performance to the k-NN algorithm may increase its performance. Using larger, more diverse datasets may also increase generalization. The Manhattan distance and cosine distance may assist explain how the distance metric affects the algorithm's performance.

### References

- Agrawal, U., Soria, D., Wagner, C., Garibaldi, J., Ellis, I. O., Bartlett, J. M. S., Cameron, D., Rakha, E. A., & Green, A. R. (2019). Combining clustering and classification ensembles: A novel pipeline to identify breast cancer profiles. *Artificial Intelligence in Medicine*, 97, 27–37. <https://doi.org/10.1016/j.artmed.2019.05.002>
- Charitopoulos, A., Rangoussi, M., & Koulouriotis, D. (2020). On the Use of Soft Computing Methods in Educational Data Mining and Learning Analytics Research: a Review of Years 2010–2018. *International Journal of Artificial Intelligence in Education*, 30(3), 371–430. <https://doi.org/10.1007/s40593-020-00200-8>
- Chichernea, V. (2014). THE USE OF DECISION SUPPORT SYSTEMS (DSS) IN SMART CITY PLANNING AND MANAGEMENT. *Journal of Information Systems & Operations Management*, 1–14.
- El-Sourani, N., Hauke, S., & Borschbach, M. (2010). An evolutionary approach for solving the Rubik's cube incorporating exact methods. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6024 LNCS(PART 1), 80–89. [https://doi.org/10.1007/978-3-642-12239-2\\_9](https://doi.org/10.1007/978-3-642-12239-2_9)
- Eremia, M., Tomsovic, K., & Cărtină, G. (2016). Expert Systems. In *Advanced Solutions in Power Systems: HVDC, FACTS, and AI Techniques*. <https://doi.org/10.1002/9781119175391.ch15>
- Gambhir, S., Malik, S. K., & Kumar, Y. (2017). PSO-ANN based diagnostic model for the early detection of dengue disease. *New Horizons in Translational Medicine*, 4(1–4), 1–8. <https://doi.org/10.1016/j.nhtm.2017.10.001>
- Giachos, I., Papakitsos, E. C., & Chorozioglou, G. (2017). Exploring natural language understanding in robotic interfaces. *International Journal of Advances in Intelligent Informatics*, 3(1), 10–19. <https://doi.org/10.26555/ijain.v3i1.81>
- Hermawan, G. (2012). “Implementasi Algoritma Greedy Best First Search pada Aplikasi Permainan Congklak untuk Optimasi Pemilihan Lubang dengan Pola Berfikir Dinamis.” *Seminar Nasional Teknologi Informasi Dan Multimedia (SNASTIA)*, 1–6. <https://doi.org/10.13140/RG.2.1.1742.4801>

- Kaushik, P., Sharma, A., & Bhathal, G. S. (2021). Revamping Supermarkets with AI and RSSi\*. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*. <https://doi.org/10.1109/ICAIS50930.2021.9395814>
- Langley, P., & Sage, S. (1994). Induction of Selective Bayesian Classifiers. *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence, 1990*, 399–406. <https://doi.org/10.1016/B978-1-55860-332-5.50055-9>
- Lumenta, A. S. M. (2014). PERBANDINGAN METODE PENCARIAN DEPTH-FIRST SEARCH, BREADTH-FIRST SEARCH DAN BEST-FIRST SEARCH PADA PERMAINAN 8-PUZZLE. *E-Journal Teknik Elektro Dan Komputer*.
- Melanie, M. (1996). An introduction to genetic algorithms. *Cambridge, Massachusetts London, England, ...*, 162. [https://doi.org/10.1016/S0898-1221\(96\)90227-8](https://doi.org/10.1016/S0898-1221(96)90227-8)
- Mutiara, I. dan A. (2015). Penerapan K-Optimal Pada Algoritma Knn Untuk Prediksi Kelulusan Tepat Waktu Mahasiswa Program Studi Ilmu Komputer Fmipa Unlam Berdasarkan Ip Sampai Dengan Semester 4. *Klik - Kumpulan Jurnal Ilmu Komputer*, 2(2), 159–173. <https://doi.org/10.20527/KLIK.V2I2.26>
- Santoso, A., Arif, I., & Hatta, M. (2017). Pembelajaran Supervised SVM Untuk Identifikasi Obyek Pisau Pada Mesin X-Ray Bandara Juanda. *NJCA (Nusantara Journal of Computers and Its Applications)*, 1(1).
- Saygili, A. (2022). Computer-Aided Detection of COVID-19 from CT Images Based on Gaussian Mixture Model and Kernel Support Vector Machines Classifier. *Arabian Journal for Science and Engineering*, 47(2), 2435–2453. <https://doi.org/10.1007/s13369-021-06240-z>
- Setiawan. (2015). Integrasi Metode Sample Bootstrapping dan Weighted Principal Component Analysis untuk Meningkatkan Performa K Nearest Neighbor pada Dataset Besar. *Journal of Intelligent Systems*, 1(2), 76–81.
- Tan, Y., Zhao, G., Dai, H., Lin, Z., & Cai, G. (2018). Classification of Parkinsonian Rigidity Using AdaBoost with Decision Stumps. *2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018*, 170–175. <https://doi.org/10.1109/ROBIO.2018.8665303>