AKADÉMIAI KIADÓ

Predicting students' academic performance using a modified kNN algorithm

Moohanad Jawthari[1*] and Veronika Stoffová[1,2]

[1] Eötvös Loránd University, Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary

[2] Trnava University in Trnava, Hornopotočná 23, 918 43 Trnava, Slovakia

Pollack Periodica • An International Journal for Engineering and Information Sciences

16 (2021) 3, 20–26

DOI: 10.1556/606.2021.00374

Received: December 31, 2020 • Revised manuscript received: April 18, 2021 • Accepted: May 13, 2021
Published online: August 7, 2021

ORIGINAL RESEARCH PAPER

ABSTRACT

The target (dependent) variable is often influenced not only by ratio scale variables, but also by qualitative (nominal scale) variables in classification analysis. Majority of machine learning techniques accept only numerical inputs. Hence, it is necessary to encode these categorical variables into numerical values using encoding techniques. If the variable does not have relation or order between its values, assigning numbers will mislead the machine learning techniques. This paper presents a modified k-nearest-neighbors algorithm that calculates the distances values of categorical (nominal) variables without encoding them. A student's academic performance dataset is used for testing the enhanced algorithm. It shows that the proposed algorithm outperforms standard one that needs nominal variables encoding to calculate the distance between the nominal variables. The results show the proposed algorithm preforms 14% better than standard one in accuracy, and it is not sensitive to outliers.

KEYWORDS

E-learning, student performance, nominal data, mixed data type classification, K-nearest neighbors, distance measures

1. INTRODUCTION

Data understanding is an important step for accurate analysis. Data pre-processing is the first step needed to aid algorithms and to improve efficiency before proceeding to the actual analysis. Data variables generally fall into one of the four broad categories: nominal scale, ordinal scale, interval scale, and ratio scale [1]. Nominal values have no quantitative value. They represent categories or classifications. For example, gender nominal variable in the datasets which take (male, female). Another one is the marital status, which takes values like (married, unmarried, divorced, and separated); here, both examples simply denote categories [1]. Ordinal variables refer to variables that show the order in measurement. For example, low/medium/high values of size variable. The ordering exists in those variables, but distances between the categories cannot be quantified. Interval scales provide order information. Besides, they possess equal intervals. For instance, the temperature is an interval data type that is measured either by Fahrenheit or by Celsius scale. Ratio scale possesses qualities of nominal, ordinal and interval scales, also has absolute zero value. In addition to, it also permits comparisons between different variables values.

The k-Nearest Neighbors (kNN) is a straightforward algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). It is a nonparametric technique that has been used in statistical estimation and pattern recognition since 1970. It uses a majority vote principle to classify new cases. Most data mining techniques cannot handle categorical variables unless they are converted to numerical variables. For example, the dataset [2] has a mixed type of attributes, categorical, and numerical. A pre-

*Corresponding author.
E-mail: moohrash@yahoo.com

AKJournals

processing step is needed to transform categorical attributes into numerical ones. There are many techniques to handle categorical values like mapping and labels encoding into Pandas, Python. However, assigning numerical values to nominal attributes misleads the machine learning algorithms learning by making difference or order between values that are not originally existed in the attributes and this phenomenon is called subjectivity. For instance, gender attribute; male can be encoded as 1 and female as 0, or the opposite. There is no standard way in encoding nominal variables.

Jawthari et al. [3] studied the effect of subjectivity where was emphasized in assigning numerical values to non-ordinal categorical attributes. That research shed the light on subjectivity using an educational dataset, especially a student performance prediction dataset. This research proposes two similarity measures for kNN algorithm to deal with categorical variables without converting them as numerical. Therefore, the algorithm overcomes the subjective encoding issue.

2. RELATED WORKS

The Educational Data Mining (EDM) is an evolving discipline that deals with the creation of methods for exploring the specific and increasingly large-scale knowledge that comes from educational environments and using these methods to better understand students and the environments in which they learn [3, 4]. One concern of EDM is predicting students' performances. The previous work [3] used various Machine Learning (ML) techniques to predict the students' performances. This article also focused on the effect of the way of encoding the nominal variables on classification accuracy of machine learning techniques as in [3], which showed that the accuracy was affected by the approach of encoding. That study, [3] recommended solving the problem using some method that does not need to convert nominal attribute to numeric. Hence, this study is to find a solution for that issue.

The kNN is one of the most popular classification algorithms due to its simplicity [5]. It stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). It classifies a new sample by a majority vote of its neighbors, with the case being assigned to the group most common amongst its k nearest neighbors kNN measured by a distance function. Euclidean distance, formula 1, is a usual similarity measure used by kNN, especially for continuous attributes, it depends mainly on the value of k. The following figure shows how the k values affect the class assignment. For instance, in Fig. 1 * refers to new point to be classified either dark square label or empty circle label. Here, * belongs to the dark square class if $k = 1$; if $k = 5$, then it is classified as the small circle class due to majority vote rule [6, 7].

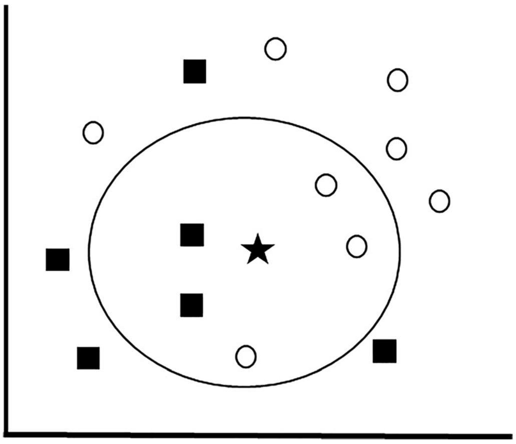


Fig. 1. The kNN classification

2.1. Distance functions

To measure the distance between points X and Y in a feature space, various distance functions have been used in the literature, in which the Euclidean distance function, Eq. (1), is the most widely used [8]. Other functions, Eqs. (3) and (4), are used to calculate the distance between continuous variables too. For categorical variables, the Hamming distance, Eq. (2), is used. Equation (3) is used to find the distance between two sets A and B and is employed as a function to find the distance between categorical variables. Let X and Y are represented by feature vectors $\mathbf{X} = \{x_1, x_2, \cdots, x_m\}$ and $\mathbf{Y} = \{y_1, y_2, \cdots, y_m\}$, where m is the dimensionality of the feature space,

$$\mathrm{d}(x,y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}, \tag{1}$$

$$D_H = \sum_{i=1}^{k} |x_i - y_i|, \quad \begin{array}{l} x = y \Rightarrow D = 0, \\ x \neq y \Rightarrow D = 1, \end{array} \tag{2}$$

$$d_J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}, \tag{3}$$

$$\mathrm{d}(x, y) = \sum_{i=1}^{k} |x_i - yi|, \tag{4}$$

$$\mathrm{d}(x, y) = \left[\sum_{i=1}^{k} (|x_i - yi|)^q \right]^{1/q}. \tag{5}$$

The k-prototypes algorithm combines the k-means and k-modes algorithms to deal with the mixed data types [9]. The k-prototypes algorithm is more useful practically because the real-world data is mixed. Assume a set of n objects, $D = \{X_1, X_2, \cdots, X_n\}$. Each X_i is called a sample or a row that consists of m attributes: $\mathbf{X}_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$.

The sample consists of numerical and categorical attributes (mn is numerical attributes, mc is categorical attributes). The aim of this algorithm is to partition the n samples into k disjoint clusters $\mathbf{C} = \{C_1, C_2, \cdots, C_k\}$, where C_i represents an i-th cluster center. K-prototypes calculate the distances between numerical features and categorical features separately and merge results.

kNN predicts the class label by majority voting of its nearest neighbors k. Let $\mathbf{X} = (x_i, y_i)_{i=1}^{N}$, where x_i is a feature vector, which has m dimensions, and y_i is the corresponding label. Given a new point or a query x_q. Its unknown label can be predicted using two steps. First step uses Eq. (1) to identify a set of k similar neighbors. Denotes the set $\mathbf{N} = \{(x_i^{NN}, y_i^{NN})\}_{i=1}^{k}$, arranged in decreasing order according to Euclidean distance. The $\delta(y = y_i)$ in Eq. (6) takes one if $y = y_i$ and zero otherwise, where y represents a class label and y_i represents the class label for the i-th nearest neighbor among its k nearest neighbors [10]. Although the majority vote is simple, it has two drawbacks: ties are possible and, all distances are equally weighted. To overcome those issues, a weighted voting method for kNN called the distance-Weighted k-Nearest Neighbor rule (WkNN) was first introduced by Dudani [11]. In WkNN, the closest neighbors are weighted more heavily than the farther ones, using the distance-weighted function. In this paper, Eq. (7) is used as a distance-weighted function to obtain the weight w_i for i-th nearest neighbor of the query. Equation (8) is used for voting to predict the new point label,

$$\widehat{y} = \arg\max_y \sum_{i=1}^{k} \delta\left(y = y_i\right), \tag{6}$$

$$w_i = \frac{1}{d\left(x_q, x_i\right)^2 + 1}. \tag{7}$$

$$F(x_q) = \arg\ \max_y \sum_{i=1}^{k} w_i \delta\left(y = y_i\right). \tag{8}$$

Here harmonic series, Eq. (9) is also used as a vote rule, which uses the rank of the k-nearest distances $(1, 2, \cdots, k)$ instead of the distances themselves, to assign weights, and compared its accuracy results with results obtained by weighted vote rule,

$$\sum_{i=0}^{k} \frac{1}{i+1} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}, \text{ harmonic series.} \tag{9}$$

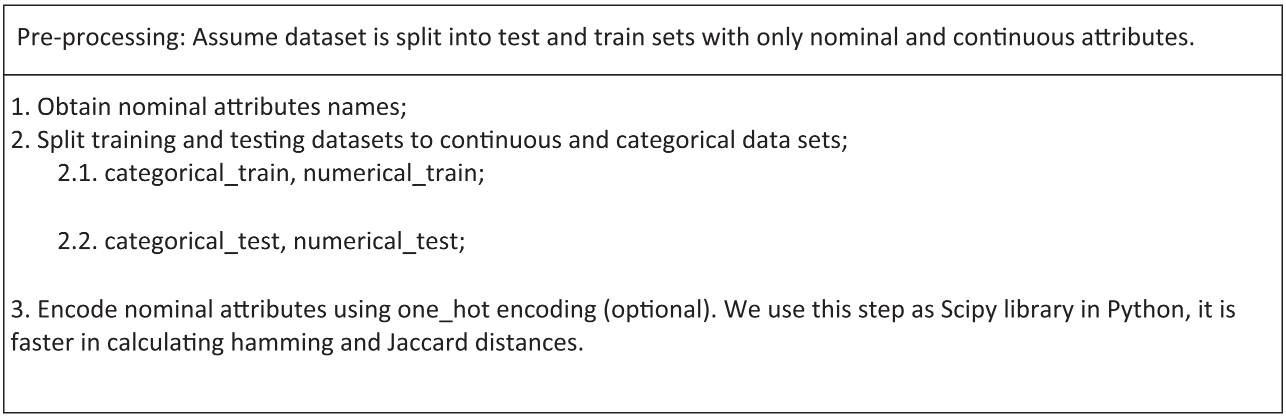
The literature is rich in methods used for clustering mixed-type datasets, but to the best of author's knowledge, there is no method that classifies categorical data using the proposed similarity measures in this paper. The proposed idea of this study was inspired by k-prototypes. Hamming distance and Jaccard distance functions are employed to obtain the distance between nominal attributes. Besides, Euclidean distance is utilized to calculate the distance of the numerical attributes as usual. The final distance is obtained by combining distance of nominal variables and distance of numerical variables.

3. PROPOSED KNN ALGORITHM

Motivated by K-prototypes mentioned above, the issue of subjective encoding of nominal variables, the issue of majority vote, a simple and effective kNN method is designed by proposing two similarity measures. The method does not need nominal variables encoding. In addition, the enhanced method considers the distance weight vote rule that give a greater weight to the closer neighbors.

The algorithm is described as below:

Algorithm 1



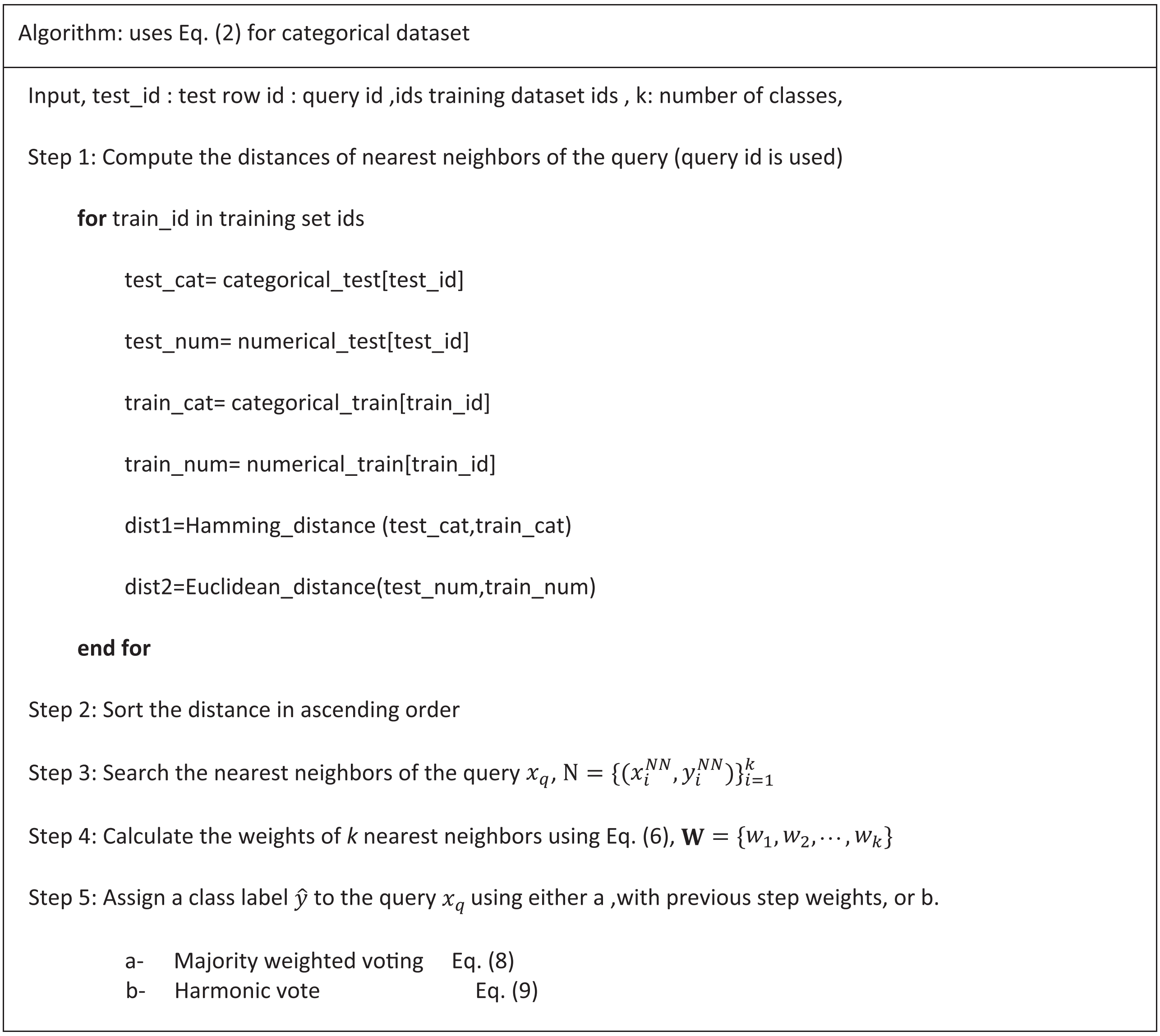
Pre-processing: Assume dataset is split into test and train sets with only nominal and continuous attributes.

```
1. Obtain nominal attributes names;
2. Split training and testing datasets to continuous and categorical data sets;
    2.1. categorical_train, numerical_train;

    2.2. categorical_test, numerical_test;

3. Encode nominal attributes using one_hot encoding (optional). We use this step as Scipy library in Python, it is
faster in calculating hamming and Jaccard distances.
```

Algorithm 2



```
Algorithm: uses Eq. (2) for categorical dataset

Input, test_id : test row id : query id ,ids training dataset ids , k: number of classes,
Step 1: Compute the distances of nearest neighbors of the query (query id is used)
    for train_id in training set ids
        test_cat= categorical_test[test_id]
        test_num= numerical_test[test_id]
        train_cat= categorical_train[train_id]
        train_num= numerical_train[train_id]
        dist1=Hamming_distance (test_cat,train_cat)
        dist2=Euclidean_distance(test_num,train_num)
    end for
Step 2: Sort the distance in ascending order
Step 3: Search the nearest neighbors of the query $x_q$, $\mathrm{N} = \{(x_i^{NN}, y_i^{NN})\}_{i=1}^{k}$
Step 4: Calculate the weights of $k$ nearest neighbors using Eq. (6), $\mathbf{W} = \{w_1, w_2, \cdots, w_k\}$
Step 5: Assign a class label $\hat{y}$ to the query $x_q$ using either a ,with previous step weights, or b.
        a-   Majority weighted voting    Eq. (8)
        b-   Harmonic vote               Eq. (9)
```

The other version of this algorithm uses Jaccard distance, Eq. (3) to calculate the similarity between categorical attributes. Besides, it used Euclidean distance for numeric attributes. Steps 2, 3, 4, and 5 above are same for the algorithm.

4. DATA SET

The data set was collected by using a learner activity tracker tool, which called experience API (xAPI). The purpose was to monitor the students' behavior to evaluate the features that may impact their performance [2].

4.1. Data mining

The dataset includes 480 student records with 16 features as it is shown in Table 1. The features are classified into three categories:

1. Demographic features such as nationality, gender, place of birth, and relation (parent responsible for student, i. e. father or mom);
2. Academic background features as educational stage, grade level, section id, semester, topic, and student absence days;
3. Behavioral features such as raised hand on class, visited resources, answering survey by parents, and school satisfaction. The dataset features are explained below:

Table 1. Dataset description

Feature	Explanation
Gender	student's gender (nominal: Male or Female)
Nationality	student's nationality (nominal: Kuwait, Lebanon, Egypt, Saudi Arabia, USA, Jordan, Venezuela, Iran, Tunis, Morocco, Syria, Palestine, Iraq, Lybia)
Place of birth	student's Place of birth (nominal: Kuwait, Lebanon, Egypt, Saudi Arabia, USA, Jordan, Venezuela, Iran, Tunis, Morocco, Syria, Palestine, Iraq, Lybia)
Educational Stages	educational level the student belongs (nominal: lowerlevel, Middle School, High School)
Grade Levels	grade student belongs (nominal: G-01, G-02, G-03, G-04, G-05, G-06, G-07, G-08, G-09, G-10, G-11, G-12)
Section ID	classroom student belongs (nominal: A, B, C)
Topic	course topic (nominal: English, Spanish, French, Arabic, IT, Math, Chemistry, Biology, Science, History, Quran, Geology)
Semester	school year semester (nominal: First, Second)
Relation	Parent responsible for student (nominal: mom, father)
Raised hand	how many times the student raises his/her hand on classroom (numeric: 0–100)
Visited resources	how many times the student visits a course content (numeric: 0–100)
Viewing announcements	how many times the student checks the new announcements (numeric: 0–100)
Discussion groups	how many times the student participate on discussion groups (numeric: 0–100)
Parent answering survey	parent answered the surveys which are provided from school or not (nominal: Yes, No)
Parent school satisfaction	the Degree of parent satisfaction from school (nominal: Yes, No)
Student absence days	the number of absence days for each student (nominal: above-7, under-7)

Figure 2 shows relationship between class variables and numerical features. It also shows the importance of behavior features. The student's who participated more in the VisiTedResourses, AnnouncementViews, RaisedHands, and Discussion-they achieved better results.

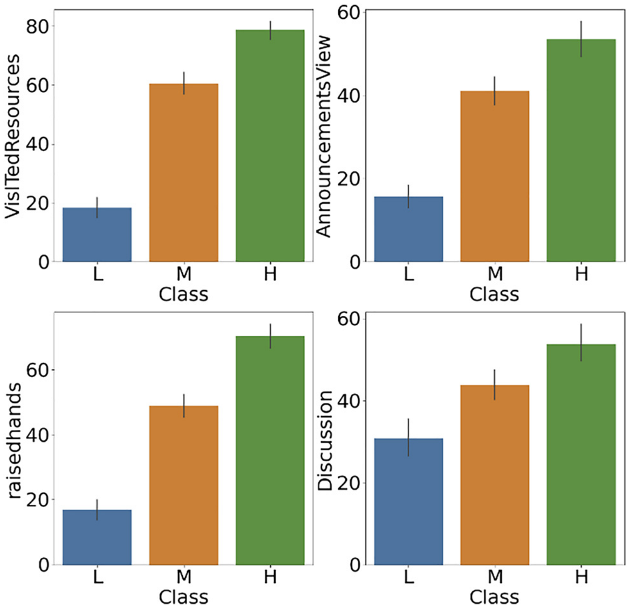


Fig. 2. Correlation between the dependent variable and numerical variables

5. RESULT AND ANALYSIS

First, the dataset is split into training and testing sets by a 20% ratio arbitrarily. The training and test datasets were split into corresponding datasets that have numerical attributes and categorical attributes. The enhanced kNN with proposed similarity measures was applied. To compare the performance of the proposed method, the categorical variables of the dataset were one_hot encoded. Then, a standard kNN from the Scikit-learn library was used [12].

The best accuracy resulted from standard kNN was 66.7 with k equals 1 as it is shown in Table 2 column 1. Hamming distance and harmonic vote kNN's best result was 72.9 with k equals 18–20 as it can be seen in Table 2, column 2. Hamming distance and weight distance vote version of kNN got the best accuracy as 78.1 with k equals 12 and 20. Columns 4 and 5 show the accuracy results of the kNN method using Jaccard distance for nominal variables. The Jaccard and harmonic vote kNN resulted in 76.0 with k equals 6 and 19 as it is described in column 4 form Table 2. The best version was the one that used Jaccard and wight distance vote as it is shown in the last column of Table 2. This version had 80.2 accuracy resulted from k equals 6. By running the algorithms multiple times, the standard one had different accuracy results. For example, one time, it had 77.0 accuracy with k equals 1. On the other hand, the proposed method had almost the same results in each run. Therefore, the proposed method is not sensitive to outliers in the data. Consequently, the proposed kNN algorithm outperforms standard kNN in accuracy. Figures 3 and 4 show the

Table 2. Accuracy results of methods

K	Standard KNN	Hamming Harmonic vote	Hamming weight distance vote	Jaccard Harmonic vote	Jaccard Weight distance vote
1	66.7	63.5	63.5	72.9	72.9
2	62.5	63.5	63.5	72.9	72.9
3	62.5	63.5	72.9	72.9	74.0
4	59.4	70.8	72.9	74.0	75.0
5	60.4	67.7	74.0	74.0	79.2
6	58.3	69.8	72.9	76.0	80.2
7	59.4	70.8	72.9	75.0	78.1
8	59.4	69.8	74.0	76.0	77.1
9	56.3	70.8	72.9	75.0	76.0
10	62.5	70.8	77.1	75.0	76.0
11	58.3	70.8	76.0	75.0	72.9
12	61.5	70.8	78.1	72.9	76.0
13	61.5	70.8	75.0	72.9	74.0
14	60.4	70.8	76.0	74.0	76.0
15	58.3	70.8	75.0	75.0	70.8
16	60.4	70.8	76.0	75.0	71.9
17	58.3	71.9	74.0	75.0	71.9
18	60.4	72.9	75.0	75.0	74.0
19	63.5	72.9	76.0	76.0	67.7
20	64.6	72.9	78.1	75.0	72.9

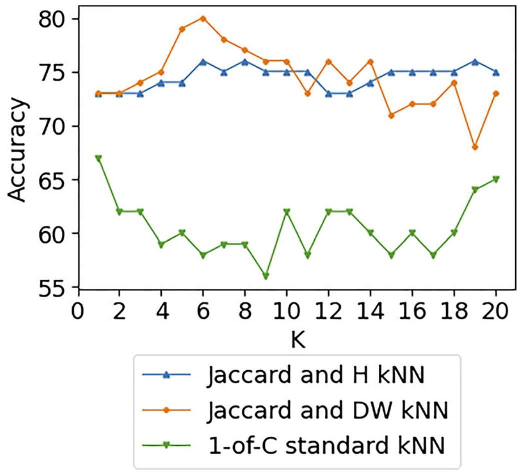


Fig. 3. Jaccard kNN results

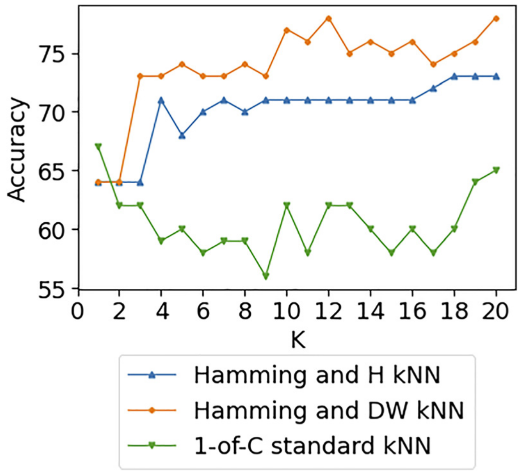


Fig. 4. Hamming kNN results

accuracy of the proposed methods with k in range between 1–20. The same figures also show the accuracy results of standard kNN that was supplied one-hot encoded nominal variables. Accuracy results were rounded up to 2 numbers.

6. CONCLUSION

This paper introduces two similarity measures to make kNN work with mixed type data, especially nominal case. The proposed method also enhanced sensitivity of kNN to outliers by considering alternative voting rules. This research contribution is to design a distance function for making classification decision without converting nominal variables to numeric. To verify the proposed classifier, experiments were conducted on the educational dataset and the results were compared with kNN algorithm after one_hot encoding nominal attributes. Experiments showed the proposed method using Jaccard distance always outperformed the standard kNN with 14%.

The enhanced kNN algorithm showed good performance in terms of accuracy, but it was slow compared to scikit-learn kNN. In the future work, the algorithm speed will be improved by incorporating fast comparing techniques. In addition, the algorithm can be used with different datasets from different fields to further show its performance.

REFERENCES

[1] K. Potdar, T. S. Pardawala, and C. Pai, “A comparative study of categorical variable encoding techniques for neural network classifiers,” *Int. J. Comput. Appl.*, vol. 175, pp. 7–9, 2017.

[2] E. A. Amrieh, T. Hamtini, and I. Aljarah, “Preprocessing and analyzing educational data set using X-API for improving student’s performance,” in *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies*, Amman, Jordan, Nov. 3–5, 2015, 2015, pp. 1–5.

[3] M. Jawthari and V. Stoffova, “Effect of encoding categorical data on student’s academic performance using data mining methods,” in *The 16th International Scientific Conference eLearning and Software for Education*, Bucharest, Romania, Apr. 23–24, 2020, 2020, pp. 521–526.

[4] P. Cortez, and A. M. G. Silva, “Using data mining to predict secondary school student performance”, in *Proceedings of 5th Annual Future Business Technology Conference*, Porto, Portugal, Apr. 1–3, 2008, A. Brito and J. Teixeira, Eds, 2008, pp. 5–12.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2007.

[6] L. Y. Hu, M. W. Huang, S. W. Ke, and C. F. Tsai, “The distance function effect on k-nearest neighbor classification for medical datasets,” *SpringerPlus*, vol. 5, Paper no. 1304, 2016.

[7] V. S. Kumar, S. A. Sivaprakasam, R. Naganathan, and S. Kavitha, “Fast K-means technique for hyper-spectral image segmentation by multiband reduction,” *Pollack Period.*, vol. 14, no. 3, pp. 201–212, 2019.

[8] D. Nagy, T. Mihálydeák, and L. Aszalós, "Graph approximation on similarity based rough sets,” *Pollack Period.*, vol. 15, no. 2, pp. 25–36, 2020.

[9] Z. Huang, “Clustering large data sets with mixed numeric and categorical values,” in *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Singapore, Feb. 23, 1997, 1997, pp. 21–34.

[10] L. Yang and R. Jin, “Distance metric learning: A comprehensive survey,” Tech. Rep., Department of Computer Science and Engineering, Michigan State University, 2006.

[11] S. A. Dudani, “The distance-weighted k-nearest-neighbor rule,” *IEEE Trans. Syst. Man, Cybernetics,* vol. 6, no. 4, pp. 325–327, 1976.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, “Scikit-learn: Machine learning in Python,” *J. Machine Learn. Res.*, vol. 12, pp. 2825–2830, 2011.