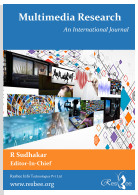
Vol.5 No.1 Jan. 2024

Multimedia Research

Received 10 September, Revised 26 November, Accepted 20 December




# Similarity Learning on Big Data: A Case Study

**Albert Agisha Ntwali**

*Official University of Bukavu, Mining School, Bukavu.*

**Abstract:** The current article aims to analyze student performance using some similarity measures. The analysis will result in a classification of the student based on how they usually take their lunch. Throughout the processes, we define some notions of similarity measures and finally select some measures to evaluate various data types of attributes. The Nearest-Neighbor approach is used for classification, with the K-Nearest-Neighbor (KNN) algorithm. At last we compare the performance on three data types: numerical, categorical and mixed data. Finally, the result is tested and validated using the Python programming language.

**Keywords:** Classifier, KNN, Similarity Measure, Student's Performance

## 1. Introduction

In the real world, one will always try to find an answer to the question of whether two given objects are similar, or whether an object $O_1$ is more similar to an object $O_2$ compared to an object $O_3$. This question seems to be a problem that many domains would like to solve. In health, a medical doctor will prescribe medicine to the patient based on case similarity. He/She will try to answer the question, is this disease similar to the one seen before? Are vital organs similar to ones seen before? So, to make a decision, the notion of similarity appears in any manner. In education, when evaluating students' results or performance, to minimize time, the committee will decide sometimes for similar cases, this will help for time optimality and for objectivity in decisions. In agriculture, crops are varied based on the similarity of the seasons. And for many other domains, we can find how they are related to similarity. Actually, we will realize that similarity is the main problem in many domains.

Now, to solve this problem, people dealing with data and models, namely data scientists, will try in the way to find a solution by training models for this task. Thus, for the prediction, analysis, and treatment of data they developed the notion of Similarity Learning. Similarity learning is defined as the process of determining a function, $s(o_1,o_2)$ which finds the optimal relation between two different data items $O_1$ and $O_2$ in a quantitative way [3].

Dealing with similarities, we will present in this work some uses of similarity measures by considering some case studies and that will be based on the types of data that we have. In the first part, we will investigate the theory of similarity learning according to the previous works done in the domain of data mining, and in the second part, we will choose some measures for each type of data and elaborate on some case studies. Finally, we will provide a classifier and incorporate the similarity measures so that we will be able to compare the performance of the classifier on different data types and using different measures. And we will try to use the classifiers to analyze a problem of students' performance on exams based on similarity learning. We will see from the analysis the classification of students based on Gender, Preparation of the test, performance on exams, and regularity in taking lunch.

## 2. Similarity Measures and Metrics

An important role is played by similarity in many machine learning problems such as classification, clustering, or ranking. For these problems, researchers built functions that could determine the similarity between attributes or objects in all the tasks[4]. These functions are tedious for real problems


https://doi.org/10.46253/j.mr.v5i1.a1

and difficult to compute manually, many works have gone into learning from labeled data to similarity and metric learning[4].

Let consider that $X$ and $Y$, two data objects, with the form :

$$x = (x_1, x_2, \cdots, x_k) \text{ and } y = (y_1, y_2, \cdots, y_k)$$

Where $k$ is the dimensionality, and each $x_i, y_i \, (\text{for} 1 \le i \le k)$, is a feature of the corresponding object. The features in our work will be of two types, either categorical or numerical. And the analysis will be made on each data type individually and on mixed features. If we talk about patterns, then the concepts of similarity and metric are reciprocal [5]. When we use similarity measures for patterns, we try to quantify the likeness between them. Also, when comparing patterns, it is very useful if they are represented in a metric space. For any set of elements characterized by the distance function between all pairs of elements, this property is important [5]. When we choose a distance *d*, d(x,y) has to follow some conditions, namely: $d(x,y) \ge 0$, where equality holds if and only if $x = y$, $d(x,y) = d(y,x)$ symmetry; $d(x,y) \le d(x,z) + d(z,y)$ triangle inequality. Note that in this article, for any distance that will be used, we will suppose that the conditions are verified and we will not have to prove that. The exception is made for thecosine similarity. It is not really a distance according to our definition.

## 2.1 Categorical Data

We consider objects whose features are categorical. Categorical data poses problems concerning similarity measures because generally, it is not possible to define a metric spacewith an implicit distance function when considering such type of data[5]. Some measures used for such type of data:

### *2.1.1 Simple Matching*

Simple matching is the simplest of all similarity measures (Van Rijsbergen 1979, [5]).It is defined as

$$SM(X, Y) = n(X \cap Y) \tag{1}$$

Where $n(A)$is just the number of elements in set *A*. $S_M$ does not take into account thesizes of each set.

### *2.1.2 Hamming*

The Hamming distance, originally, defined for binary codes can be applied to anyordered sets of equal length. This measure is defined by

$$d_H = \sum_{i=1}^{n} \delta(x_i, y_i) \tag{2}$$

Where

$$\delta = \begin{cases} 1, & \text{if } x_i = y_i \\ 0, & \text{if } x_i \ne y_i \end{cases}$$

If the features are binary the Hamming distance, simple matching measure, and the squared Euclidean distance (to be defined), become equivalent [5]. Many similarity measures are used for binary valued features. Thus, to use them we have to convert categorical features into binary features.

### *2.1.3 Cosine Similarity*

For the categorical data, we can also use the cosine similarity. Between two documentsfor example we can find the cosine similarity between two documents. The cosine measure is defined by

$$\text{Cos}\,(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{n} x_i, y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3}$$

The cosine measure ignores the relative frequencies and will require the transformation of data as in binary or in the numerical label.

### *2.1.4 Jaccard Similarity*

The Jaccard similarity of two sets $X$ and $Y$ , said Jaccard coefficient, is defined as the ratio between the size of the intersection of the two sets and the size of the reunion. We denote and define the Jaccard index (or coefficient) by

$$J_S(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{4}$$

And when we have binary vectors, we can define it as

$$J_s = \frac{\sum_{i=1}^{d} x_i . y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i . y_i} \tag{5}$$

with $\overline{X} = (x_1, \cdots, x_d)$ , $\overline{Y} = (y_1, \cdots, y_d)$ and $x_i, y_i \in \{0,1\}$ .

Note that the Jaccard distance measures the dissimilarity between sets and it is defined by

$$J_\delta(X,Y) = 1 - J(X,Y) = \frac{|X \cup Y| - |X \cap Y|}{|X \cup Y|} \tag{6}$$

For our case study, we will use the Jaccard coefficient (5). Many other similarity measures can be used for categorical data such as Overlap, Fowlkes mallows, Mountford, etc.

## 2.2 Numerical Data

Many researchers are actually focused on categorical data because, indeed, many measures do not have complication when dealing with numerical data. For this reason, we will find that many measures are provided for numerical and actually, data scientists look at ways to adapt them to categorical features. Let us present some measures that can be used for numerical data :

### *2.2.1* $L_p$ *-Norm,* $p \geq 1$

Given $\overline{X} = (x_1, \cdots, x_n)$ and $\overline{Y} = (y_1, \cdots, y_n)$, we define the distance

$$Dist(\overline{X}, \overline{Y}) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{7}$$

At specific values of *p*, we have well-known measures.

- For $p = 1$, the Manhattan norm (distance)

$$Dist(\overline{X}, \overline{Y}) = \left( \sum_{i=1}^{n} |x_i - y_i| \right) \tag{8}$$

- For $p = 2$: Euclidean norm

$$Dist(\overline{X}, \overline{Y}) = \left( \sum_{i=1}^{n} |x_i - y_i|^2 \right)^{\frac{1}{2}} \tag{9}$$

- For $p = \infty$ : Infinity norm

$$Dist\ (\overline{X}, \overline{Y}) = \max_i |x_i - y_i| \tag{10}$$

Many things can impact the performance of these measures, namely, the dimensionality, the irrelevant features.

## 2.3 Mixed Data

In some circumstances, we have mixed data where a part of features (attributes) in the data is numerical and another part categorical. Let us try to give thinking about how we can deal with such type of dataset. Given two groups of data $\overline{X} = (\overline{X_n}, \overline{X_c})$ and $\overline{Y} = (\overline{Y_n}, \overline{Y_c})$ where $\overline{X_n}, \overline{Y_n}$ are subsets of numerical attributes and $X_c$, $Y_c$ are subsets of categorical attributes.
To find the similarity in mixed data we use a weighted average. We define it as

$$Sim(\overline{X}, \overline{Y}) = \lambda . NumSim(\overline{X_n}, \overline{Y_n}) + (1 - \lambda) . CatSim(\overline{X_c}, \overline{Y_c}) \tag{11}$$

Where *NumSim* is the similarity measure for the numerical data and *Catsim* is the similarity measure for categorical data. The function *Sim* is the new similarity measure for combined data. But, the value of the portion $\lambda$ is difficult to decide.

We can also use the **Normalized Weighted average**:

$$Sim(\overline{X},\overline{Y}) = \lambda.\frac{NumSim(\overline{X_n},\overline{Y_n})}{\sigma_n} + (1-\lambda).\frac{CatSim(\overline{X_c},\overline{Y_c})}{\sigma_c} \quad (12)$$

Where $\sigma_c$, $\sigma_n$ are the standard deviation for the categorical and the numerical data. With this thinking, we will need to know, most of the time, the distribution of the data. Another thought is that we can transform all the data in numerical, doing encoding and using a unique similarity measure for all the datasets. This second idea is the one that will be used in this article. But the consequence is that some information can be lost or you will have a huge dataset to deal with after transformation, that will not be supported by the memory of the system.

# 3. Similarity Learning

For our learning task, we will have to analyze student performance in the exam according to some attributes. We will consider for our dataset many attributes some of which are categorical and others which are numerical.

For the analysis, we will not consider all the attributes provided by the data set. We will focus on gender, test preparation, and grades in Mathematics and in Reading.

## 3.1 Mixed Data

For our case study, since we are dealing with mixed data, we will consider the Jaccard coefficient as a similarity measure for the categorical data part and the Euclidean distance $L_2$–Norm as a similarity measure for the numerical data.

For the mixed data, the mixed similarity measure as our first thiking is to use the weighted average. And for that, we denote $E-J$ and define a new measure given $X, Y$ as

$$E-J = \lambda.L_2(\overline{X_n},\overline{Y_n}) + (1-\lambda).J_S(\overline{X_c},\overline{Y_c}) \quad (13)$$

Where $L_2$ represents the Euclidean distance and $J_s$ indicates the Jaccard coefficient.

We will have to verify if this similarity measure evaluates the similarity according to our data set otherwise we redefine another expression, the purpose being looking for efficiency. In this work, due to the limited time of implementation and the purpose of the production of this report, we will use our second idea, which is to transform all the data into numerical data by encoding. Later, we will think about the implementation of the expression(13) and provide more theory related to the idea.

## 3.2 Homogeneous Data

For homogeneity, we will consider the same dataset as in the mixed case study and for each part, we will consider the data type corresponding to the case. The numerical part of the initial dataset will be used for similarity measure in the Numerical case, and the categorical part for the Categorical case.

### *3.2.1 Numerical Data*

For the Numerical Case, we will use the Euclidean measure (metric or distance) $L_2$,

$$L2(\overline{X},\overline{Y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{\frac{1}{2}} \quad (14)$$

Even though it has some limits, it is a recommended measure for the proximity process andit is easy to manipulate and has other advantages. To compute the similarity we will have to normalize the Euclidean distance. This is helpful because sometimes the directionof the vector is more meaningful than the magnitude. The normalized distance, which defines our similarity, will be defined by

$$SimNum = \frac{1}{1+L_2(\overline{X},\overline{Y})} \quad (15)$$

### *3.2.2 Categorical Data*

For the categorical case, we will use the Jaccard coefficient as a measure to evaluate the similarity. The data will be provided by the categorical part of the initial dataset. Recall, the Jaccard is defined as

$$J_s = \frac{\sum_{i=1}^{d} x_i . y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i . y_i} = \frac{|X \cap Y|}{|X \cup Y|} \quad (16)$$

This assumed we have two vectors X and Y of the same dimension. Further explanation is given by the computation and the implementation of the data.

we can also use the cosine distance (not a metric) to define the categorical data similarity since for non-binary sets it is difficult to use the Jaccard coefficient.

## 3.3. Classifiers for Mixed Data

We choose to work with the Nearest-Neighbor-based classifiers. Indeed, among the methods of supervised learning, the Nearest Neighbor rule achieves consistently high performance, without distribution assumptions at the beginning of the analysis. Also, since our problem will be using classification for prediction, we think this classifier is good for our learning choice. we will be using the KNN algorithm for the implementation and the analysis.

# 4. Experiment and Evaluation

This section describes the experimental analysis and the evaluation. The Dataset used in this work is Data: "HTTPS://WWW.KAGGLE.COM/SPSCIENTIST/STUDENTS-PERFORMANCE-IN-EXAMS".

Table 1 represents the sample of the data set used for the analysis. The student will be classified according to their lunch taken in a standard way or not taken (free) or reduced.

## 4.1 Categorical data

The extrait of the square matrix, which is represented in table 1, presents the Jaccard coefficient, between objects. Note that the size of the matrix is $100 \times 100$.

Table 2 indicates the Jaccard similarity on Categorical encoded data. Jaccard similarity (coefficient) is 0 between the same object. For some objects, we will see that this coefficient is large and close to 1.0 which means there is a similarity (relation) between students. It is with respect to the gender and the test preparation. Lunch is a categorical attribute, but it will be used for prediction.

## 4.2 Numerical data

The extrait of the square matrix, which is represented in table 5.2, presents the similarity measure, here the normalized Euclidean distance (15), between objects with numerical attributes. The Euclidean distance was computed with respect to the Math score and the Reading score. Note that the size of the matrix is $100 \times 100$. Table 3 summarizes the Euclidean measure of Numerical data. The Euclidean measure as present gives one when we find the similarity between an object and itself, in fact, the expression (15) gives one when the euclidean distance goes to zero. Looking at the table5.2, we will see that the two subjects are not related. These two attributes are independent (Math and Reading).

***Table 1:*** *Dataset sample*

| ID | gender | test preparation | math score | reading score | lunch |
|---|---|---|---|---|---|
| **1** | female | none | 72 | 72 | standard |
| **2** | female | completed | 69 | 90 | standard |
| **3** | female | none | 90 | 95 | standard |
| **4** | male | none | 47 | 57 | free/reduced |
| **5** | male | none | 76 | 78 | standard |

## 4.3 Mixed data

For the mixed data we used the Euclidean measure since we were able to convert all the data into numerical types. The process is as in the Numerical case. The extrait of the square matrix that is presented is nothing else but the normalized euclidean measure onthe mixed data.

Note that the values in Table 4 are estimations. In the code, we also provide the similarity between attributes for this case.

We can see by the mixed data that the similarity between students when considering all the attributes is very small. This means that, in this case, the performance is not relatedto the relation among students based on the difference attributes. We may be wrong according to some human realities. But by the collected data the analysis is clear, andwe do not have to forget the fact of changing data types that could bring errors.

## 4.4 Error Analysis and Evaluation

For this section, we provide the variation of errors when k changes and evaluate the classifier on different case studies. Globally, we just see the variation of the errors atdifferent k's and the accuracy at training and testing of the data.

### *4.4.1 Analysis of Numerical*

Fig 1 demonstrates the error analysis on the KNN classifier for numerical. Here, the mean error is calculated by varying the K-value.

***Table. 2:*** *The Jaccard similarity on Categorical encoded data.*

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 0.0 | 0.667 | 0.00 | 0.667 | 0.667 | 0.0 | 0.667 | 0.667 |
| 0.667 | 0.0 | 0.667 | 1.0 | 1.0 | 0.667 | 0.0 | 1.0 |
| 0.0 | 0.667 | 0.0 | 0.667 | 0.667 | 0.0 | 0.667 | 0.667 |
| 0.667 | 1.0 | 0.667 | 0.0 | 0.0 | 0.667 | 1.0 | 0.0 |
| 0.667 | 1.0 | 0.667 | 0.0 | 0.0 | 0.667 | 1.0 | 0.0 |

***Table 3:*** *Euclidean measure of Numerical data.*

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... |
|---|---|---|---|---|---|---|---|---|---|
| 1.0 | 0.052 | 0.0331 | 0.0332 | 0.1218 | 0.083 | 0.034462 | 0.022632 | 0.081 | ... |
| 0.0519 | 1.0 | 0.0443 | 0.0246 | 0.067148 | 0.120771 | 0.0484 | 0.017785 | 0.0364 | ... |
| 0.0331 | 0.0443 | 1.0 | 0.0171 | 0.043435 | 0.042604 | 0.333 | 0.013673 | 0.0241 | ... |
| 0.0332 | 0.0246 | 0.0171 | 1.0 | 0.02717 | 0.027485 | 0.017574 | 0.060051 | 0.052 | ... |
| 0.1218 | 0.0671 | 0.0434 | 0.02717 | 1.0 | 0.123899 | 0.0458 | 0.01953 | 0.051443 | ... |

***Table 4:*** *Euclidean measure on mixed data.*

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... |
|---|---|---|---|---|---|---|---|---|---|
| 1.0 | 0.0518 | 0.0331 | 0.0331 | 0.119782 | 0.08302 | 0.03441 | 0.0226 | 0.0801 | ... |
| 0.052 | 1.0 | 0.0441 | 0.0246 | 0.0665 | 0.119 | 0.0484 | 0.0177 | 0.036345 | ... |
| 0.0331 | 0.0442 | 1.0 | 0.0171 | 0.0434 | 0.0426 | 0.2899 | 0.013670 | 0.024091 | ... |
| 0.03312 | 0.0246 | 0.0171 | 1.0 | 0.0272 | 0.0275 | 0.0176 | 0.060051 | 0.051443 | ... |
| 0.1198 | 0.0665 | 0.0434 | 0.0272 | 1.0 | 0.122 | 0.0457 | 0.019 | 0.0513 | ... |

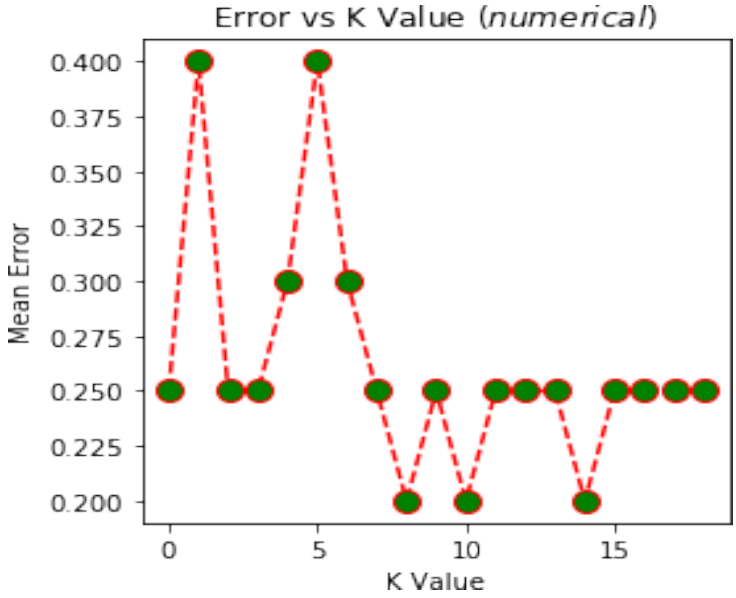



***Fig.1.*** *Error on KNN for numerical*

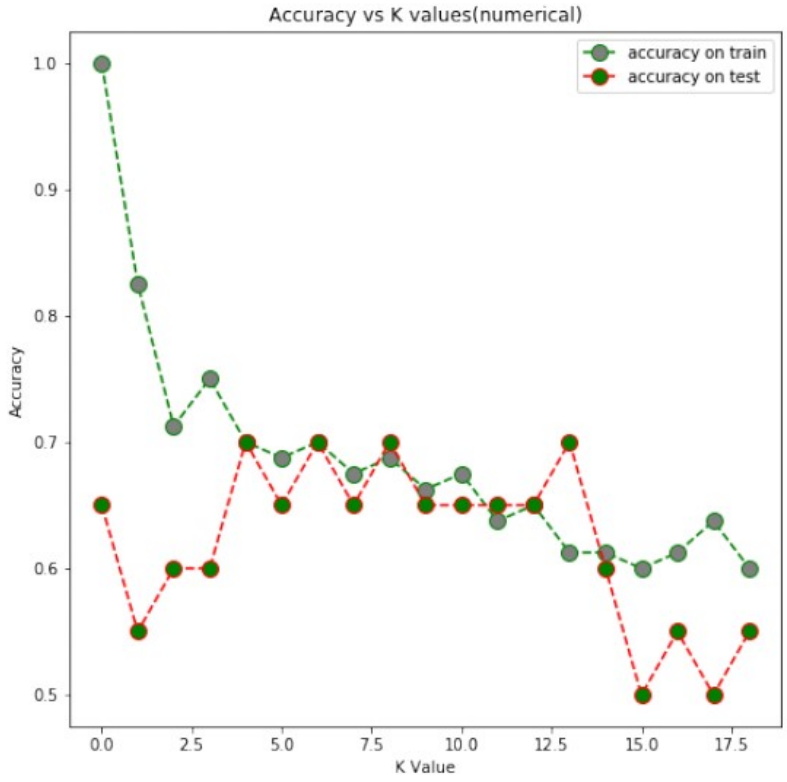



***Fig.2.*** *Accuracy of training andtesting at different k values.*

Fig 2 exhibits the accuracy of training and testing at different k values. Here, the experimental analysis is performed by varying the K value. Moreover, the accuracy of the training set is higher, wherein the accuracy on test at the final iteration.

*Table 5: Performance analysis of numerical*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **free/reduced** | 0.00 | 0.00 | 0.00 | 8 |
| **standard** | 0.60 | 1.00 | 0.75 | 12 |
| **micro avg** | 0.60 | 0.60 | 0.60 | 20 |
| **macro avg** | 0.30 | 0.50 | 0.37 | 20 |
| **weighted avg** | 0.36 | 0.60 | 0.45 | 20 |

Accuracy score on training:0.75
Accuracy score on testing:0.7

### *4.4.2 Analysis of Categorical*

Fig 3 demonstrates the error analysis of KNN for categorical. Here, the experimental analysis is performed by varying the K value and computing the mean error. When the number of iterations increases, the error gets reduced.

Fig 4 demonstrates the accuracy of training and testing at diverse k values. Here, the experimental analysis is performed by varying the K value and computing the accuracy.

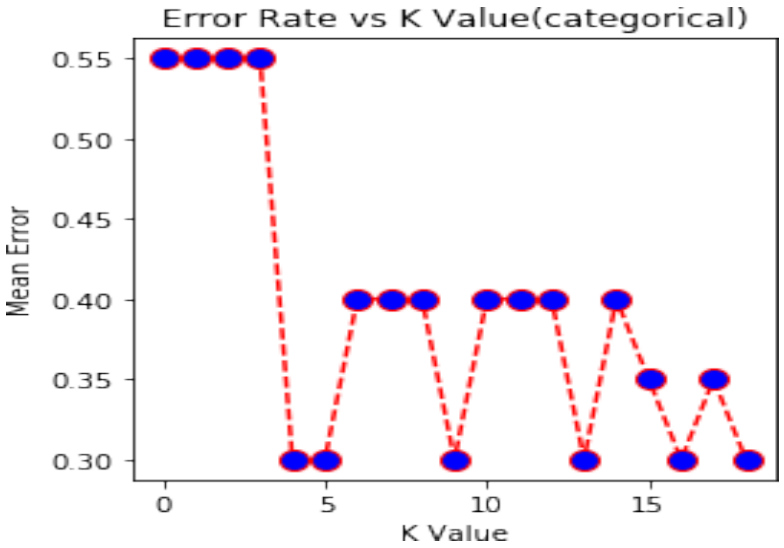



***Fig.3.****Error on KNN for Categorical*

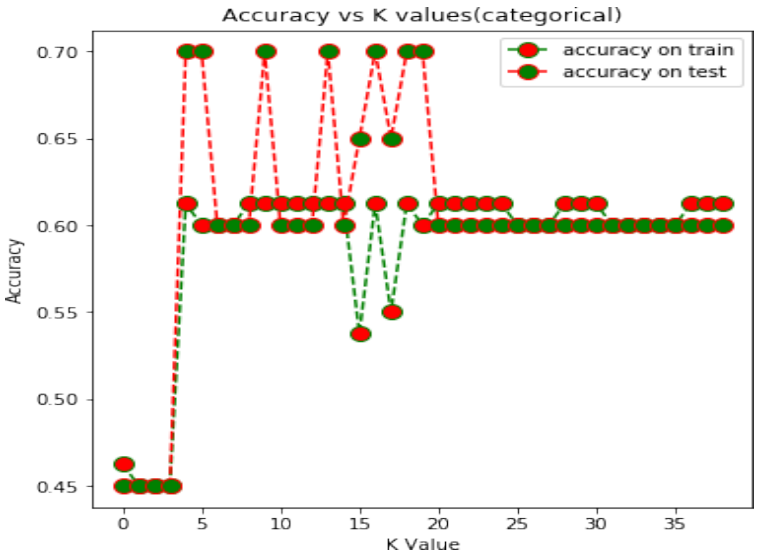



***Fig.4.*** *Accuracy of training andtesting at different k values.*

*Table 6: Performance analysis of Categorical*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **free/reduced** | 0.00 | 0.00 | 0.00 | 8 |
| **standard** | 0.60 | 1.00 | 0.75 | 12 |
| **micro avg** | 0.60 | 0.60 | 0.60 | 20 |
| **macro avg** | 0.30 | 0.50 | 0.37 | 20 |
| **weighted avg** | 0.36 | 0.60 | 0.45 | 20 |

Accuracy score on training:0.6125
Accuracy score on testing:0.6

### *4.4.3 Analysis of Mixed*

Fig 5 demonstrates the error analysis of KNN for Mixed data. Here, the experimental analysis is performed by varying the K value and computing the mean error. During the last iterations, the error gets increases. Fig 6 shows the accuracy of training and testing at diverse k values. Here, the experimental analysis is performed by varying the K value and computing the accuracy.

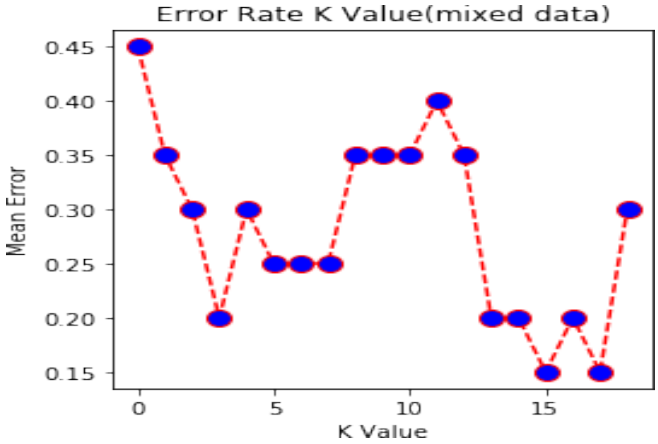



***Fig.5.*** *Error on KNN for Mixed data*

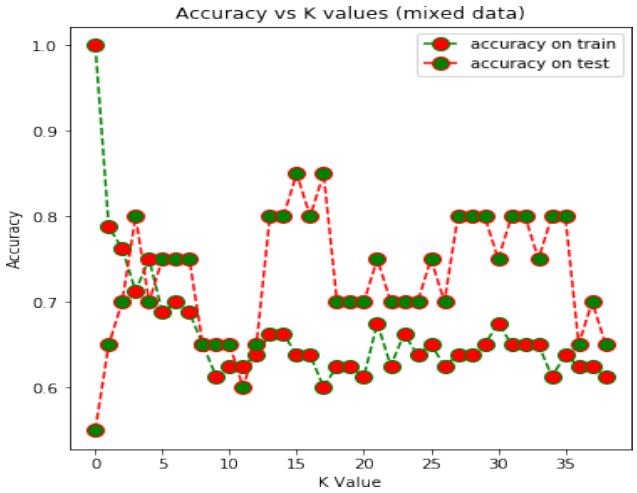



***Fig.6.*** *Accuracy of training andtesting at different k values.*

*Table 7: Performance analysis of mixed data*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **free/reduced** | 0.00 | 0.00 | 0.00 | 8 |
| **standard** | 0.60 | 1.00 | 0.75 | 12 |
| **micro avg** | 0.60 | 0.60 | 0.60 | 20 |
| **macro avg** | 0.30 | 0.50 | 0.37 | 20 |
| **weighted avg** | 0.36 | 0.60 | 0.45 | 20 |

Accuracy score on training: 0.75

Accuracy score on testing: 0.7

## 5. Conclusion

The inspiration of this project was the fact that student can perform well or bad in the exam, and that can be related sometimes to the hunger or satisfaction. For our analysis, we tried to see if the attributes are related. If there are first similarities between studentswhat we called object similarity and than tries to see if there are similarities between attributes what we called attributes similarity. Many cases are provided in the implementation. Evaluating all the case studies, we have seen that the Nearest Neighbor classifier perform well on the mixed data, with the Euclidean measure, compare to how it performs on the two other data types, using the Euclidean and the Jaccard, respectively on the numerical and the categorical. If we could add more attributes and more objects The result could totally change.

For future work, we would try to implement the EJ measure, that uses two similarity measures, namely the Euclidean distance and the Jaccard coefficient, to evaluate the similarity between mixed data types. Or to investigate other similarities that can be matched together in the weighted average-oriented approach. Also, one could variate the measures when using the same classifier and evaluate its performance. One could change the classifiers and use the same measures for evaluation.

## Compliance with Ethical Standards

**Conflicts of interest:** Authors declared that they have no conflict of interest.

**Human participants:** The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

## References


[1] Longbing Cao, *Data Science Thinking*,The Next Scientific, Technological and Eco- nomic Revolution. Springer 2018.

[2] Lijun Zhang, *Similarity and Distances*. http://cs.nju.edu.cn/zlj

[3] https://link.springer.com/referenceworkentry/10.1007% 2F978-1-4419-1428-6_1811

[4] Aurélien Bellet, Amaury Habrard, Marc Sebban, *Similarity Learning for Provably Accurate Sparse Linear Classification*. Laboratoire Hubert Curien UMR CNRS 5516, Université Jean Monnet, 42000 Saint-Etienne, France.

[5] Fernando L., Victor L., and Fernando B. , *Binary-based similarity measures for cat- egorical data and their application in Self Organizing Maps*. Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa.

[6] Yihua Chen, Eric K., Ali Rahimi, Maya R., Luca Cazzanti *Similarity-based Classifi- cation: Concepts and Algorithms*. Journal of Machine Learning Research 10 (2009) 747-776.

[7] Emilia L., Francisco G., Miguel A., *Classification similarity learning using feature-based and distance-based representations: a comparative study*. Departament d'Informàtica, Universitat de València.

[8] Ali Seyed S., Saeed A., Teh Ying Wah , *A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data*. Research article. Plos | One.

[9] Anand Rajaraman Kosmix, Jeffrey D. Ullman, *Mining of Massive Datasets*. Stanford University.

[10] Anand Rajaraman Kosmix, Jeffrey D. Ullman, *Mining of Massive Datasets*. Stanford University.

[11] Athar Kharal, *Distance and Similarity Measures for Soft Sets*. http://arxiv.org/ abs/1006.4904v1. 25 June 2010.

[12] *Euclidean Distance raw, normalized, and double-scaled coefficients*. https://www. pbarrett.net/techpapers/euclid.pdf.

[13] Shyam Boriah, Varun Chandola, Vipin Kumar, *Similarity Measures for Categorical Data: A Comparative Evaluation*. Department of Computer Science and Engineering,University of Minnesota.

[14] V. B. Surya Prasatha, Haneen Arafat Abu Alfeilat, Omar Lasassmeh, Ahmad B. A. Hassanat, *Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier – A Review*. arXiv:1708.04321v1.14Aug2017.