International Journal of Engineering &Technology, 8 (4) (2019) 523-526

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



k-NN improvement to data analysis

Tonya P. Mateva [1] *, Ivan G. Ivanov [2]

[1] College of Dobrich, Shumen University, Dobrich, Bulgaria
[2] Faculty of Economics and Business Administration, Sofia University St. Kl. Ohridski, Bulgatia
**Corresponding author E-mail: i_ivanov@feb.uni-sofia.bg*

Abstract

The problem to classify big data is an actual one the subject. There are multiple ways to classify data but the k Nearest Neighbors (k-NN) has become a popular tool for the data scientist. In this paper we examine several modifications of the k Nearest Neighbors algorithm that achieve better efficiency in terms of accuracy and CPU time when classifying test observations in comparison to the standard k Nearest Neighbors algorithm. To make the modifications faster than standard k-NN we use a special methodology which splits the input dataset into n folds and combine it with input data transformations. Each time we execute the process, one of the folds is saved as a test subset and the rest of the folds are applied for training. The process is executed n times. In the proposed methodology we are looking for the pair of subsets which produces the highest accuracy result.

Keywords: *Classification Problems; Data Analysis; K Nearest Neighbors (K-NN); Machine Learning.*

1. Introduction

Data Analysis is becoming relevant over the most recent 20 years. The task of classifying a dataset with a large number of observations is one of the major research studies on the big data analysis. The method of the k-Nearest Neighbors is to determine to which class a new observation belongs, the method finding its closest neighbors whose class is known [11]. Nearest neighbors are determined based on the distance between the new observation.

Although it is mainly used to resolve classification problems, it is not unusual for k-NN to be used for image recognition, text categorization, object recognition, and more [3], [5], [6], [9]. The data analysis tends to develop efficient computational techniques that will raise with experience, for analysis the vast complex data sets such as complex biological data [5], [8], [13]. Different approaches and algorithms Artificial neural networks, k- Nearest Neighbors, Random forest, Support vector machine have been investigated [17]. Fraud detection in healthcare domain are continuously evolving and are put into practice in many business fields. User behavior is monitored in fraud detection in order to analyze and find any suspicious or undesirable behavior and to avoid the same. Different types of fraud detection machine learning techniques are commented in [14].

The classic k-NN can be applied in customer relationship processes by more efficient filtering of prospective buyers of a particular product or service, as it may classify them as buyers or not buyers [1]. Different modifications based on the classic k-NN method have proposed in the literature [3], [4], [18].

2. Methodology

The process of classifying big data sets passes through several stages [12]. In the beginning, the set of data is prepared - this is done in two ways. The main set is divided into two subsets, called training and test. The model is constructed (the term is trained) on the training set, and the model is checked on the test subset to determine how it predicts the observations in it. The test subset is unknown to the model. In the second approach, the given set is divided into three subsets - training, test and validation. The validating subset does not change throughout the process, while the attitude towards the training and test is as in the first case. Observations may vary from the training to the test subset and vice versa. The aim is to find a model on the training subset with the best performance on the test subset. The model found is validated on the validation subset. Observations from the validation subset are new to the built model, and the properties of the built model can be evaluated.

In this paper we propose some algorithm modifications of the k Nearest Neighbors method for accessing high-quality kNN model that achieves better efficiency in terms of accuracy and F1 precision when classifying test observations in comparison to the standard kNN algorithm. To validate the results obtained from the proposed kNN modifications and to assess the performance, the predictions of them are tested on the datasets available on the Internet and compared with the standard k-NN algorithm. In this investigation we apply the Python information technology.

The estimation or verification of the predictive power of a model is done through existing criteria for the purpose. These are different criteria that have different degrees of relevance for different classification methods. The most commonly used criterion is implemented

Copyright © Tonya P. Mateva, Ivan G. Ivanov. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

through the Python middle score command and it measures the difference between the actual and predicted values of the dependent variable. This criterion can be used for all subsets in a classification task - a training, test, validation subset and the entire (total) set. But any such criterion has real strength and significance when used on observations unknown to it, that is, it is not recommended to apply to the training subset in the sense that the model is built on the same subset. Model evaluation is performed in the third stage of the model. The numeric score value is a number in the zero and one. The goal is to find a model that shows a numeric score value on the test set (test or validation) close to a unit. During the second stage, the parameters for applying the model classification method are selected to allow it to show a large predictive power after construction.

Objective of the study:

To demonstrate k-NN algorithms variety with different data preprocessing.

To promote about kNN algorithms and to describe their experimental properties.

Because we will apply the method of closest neighbors in this report, we will describe the methodology for its application in the terminology used. The information technology we will work with is the Anaconda environment with Python 2.7. The k Nearest Neighbors method builds a pattern based on some observations that are pre-classified, i.e. broken down by classes [7, 10]. The built model should predict to which class each new observation belongs, based on the classes of the nearest neighbors of the new observation. For example, if multiple observations are made up of two classes and a new observation occurs, the nearest observations are set to the new one (k is the number of neighbors and is determined in advance). The classes of these closest neighbors are examined and the new observation is classified into the predominant class. To implement the algorithm, the set of observations is divided into two subsets - a subset and a subset of the test subset. In addition, in some situations, even when the training set is "big enough," finding nearest neighbors can be done very quickly.

Table 1: The Set Diabetes Actually Contains the Data from the Pima Indians Diabetes. The Last Access to the Cited Sites was Made in April 2019

	Number of observations	Number of classes	Source
Diabetes	768	2	https://gist.github.com/ktisha/c21e73a1bd1700294ef790c56c8aec1f
German	1000	2	https://onlinecourses.science.psu.edu/stat857/node/215 https://onlinecourses.science.psu.edu/stat857/node/222
Liver	345	2	https://sci2s.ugr.es/keel/dataset.php?cod=68
Vehicle Silhouettes	846	4	https://sci2s.ugr.es/keel/dataset.php?cod=68
winequality-red	1599	6	https://github.com/zygmuntz/wine-quality/tree/master/winequality http://www3.dsi.uminho.pt/pcortez/wine/
winequality-white	4898	7	https://github.com/zygmuntz/wine-quality/tree/master/winequality http://www3.dsi.uminho.pt/pcortez/wine/

3. Algorithms

First Approach (Algorithm 1). We divide the main set by the train_test_split command to a training and test subset, and by the parameter test_size the size of the two subsets is determined. We choose the number of neighbors, ie. the value of k. We train the model on the train subset and check it on the test subset. We run the algorithm several times. In each execution, responses are different. And this is because in each execution the division of the training and test set is different. The algorithm described above is standard and is most commonly used to classify big data sets. It is found in many Internet applications and is detailed in a number of books and Internet sites.

Second Approach (Algorithm 2). The basis of this algorithm is the idea that the value of k changes, as well as the observations that form the training and test subsets. We organize a loop to realize the division of the main set as in Algorithm 1, and remember the values of k which produces the highest value of the score coefficient on the test subset. As a result, we determine the number of neighbors k* for which the model reaches the highest predicted power.

Third Approach (Algorithm 3). The main set is divided into two subsets - training and testing as in Algorithm 1. Next, we organize a loop to determine the value of the number of neighbors k. As a result, for the already divided set, we find the best value k*, in which the model reaches the highest value of the score coefficient on the test subset. We run the algorithm several times. In each execution, the responses are different because the subsets are different.

Fourth Approach (Algorithm 4). In this algorithm, we first organize a loop on the k to determine the number of neighbors. Within the loop, we divide the main set of training and testing using the idea developed and described in detail in Ivanov and Tanov's textbook [12]. Than is a different reorganizing of the data before construct the model. In short, this is achieved by the command kf = KFold (len (X), n_folds = 6, shuffle = True) which divides the base set X into n_folds equal parts (in this case n_folds = 6). The division depends on the logical variable shuffle. With the true value of this variable, the observations are randomly divided into each n_folds section. Each time the KFold command is executed at true value of the logical variable, a different division is obtained, then a different model with different predictive power. If this variable has a false value, then the observations retain their ordinance from the main set in each subset. In each subsequent execution of the KFold command, the results of the change are changed because the division of the base set does not change.

Fifth approach (Algorithm 5). The goal here is to divide the main plurality of subsets to choose the number of neighbors so that we get the highest value for the score coefficient on the test subset. In this algorithm, we begin by dividing the main set by the command kf = KFold (len (X), n_folds = 6, shuffle = True). We organize a cycle with repeated n_folds times to determine a training and test subset. Within the loop, we define an internal loop to determine the number of neighbors. As a result, we find a division of the main set and the number of neighbors for which we have a high score value.

4. Experiments and discussions

In this section we will conduct experiments to compare the performance of the four data classification algorithms. The algorithms will apply to several sets of data listed in Table 1. The data sets are pre-classified and the observations are divided into classes. With each algorithm we will build a model that we will test on the test subset, and then we can use it to classify new observations that we do not know to which class they belong. We run the experiments on the Anaconda and Python 2.7 software platform via a 1.81GHz PENTIUM® Dual CPU computer.

In order to estimate the effectiveness of the considered algorithms we compare the success rate of each, presented via score coefficient received by the command score and the F1-score coefficient from the classification report.

At the core of all algorithms is the (KNeighborsClassifier k-NN) standard procedure that is described and used in Python in the sklearn.neighbors library (see [16]). The main commands are as follows: Command to define the model (knn = KNeighborsClassifier n_neighbors = ...); the training command of the knn.fit (X_train, y_train) that applies to the training subset (X_train, y_train); a command for determining the score coefficient knn.score (X_test, y_test) applied to the test subset (X_test, y_test). The results of applying the algorithms described in Table 2 are presented. The knn.score () value reaches numerical results corresponding to the algorithms in the columns of Table 2 are described.

Table 2: The Results from the Experiments

	A1, k, score, F1-score	A2, k, score, F1-score	A3, k, score, F1-score	A4, k, score, F1-score	A5, k, score, F1-score
Diabetes	k=5, 0.79, 0.79	k=19, 0.76, 0.76	k=15, 0.75, 0.75	k=17, 0.81, 0.81	k=9, 0.82, 0.82
German	k=5, 0.72, 0.72	k=11, 0.72, 0.62	k=15, 0.74, 0.70	k=13, 0.77, 0.74	k=13, 0.78, 0.76
Liver	k=5, 0.66, 0.65	k=9, 0.74, 0.69	k=11, 0.72, 0.70	k=7, 0.82, 0.81	k=5, 0.79, 0.78
Vehicle Silhouettes	k=5, 0.66, 0.64	k=3, 0.65, 0.60	k=3, 0.67, 0.67	k=5, 0.723, 0.72	k=5, 0.73, 0.72
winequality-red	k=5, 0.49, 0.47	k=1, 0.54, 0.54	k=1, 0.57, 0.56	k=1, 0.64, 0.63	k=1, 0.63, 0.64
winequality-white	k=5, 0.48, 0.46	k=1, 0.56, 0.56	k=1, 0.55, 0.55	k=1, 0.61, 0.61	k=1, 0.60, 0.60

There are many similar investigations in the literature. For example, the liver data set has been analyzed in [2] where the kNN classifier is used for all features and using feature selection approach. The achieved accuracy in these two algorithms are 69.08% and 75.04%, respectively. The accuracy provided in [15] for classification of the liver dataset is 69.58%. Moreover, our results presented in Table 2 show that the best accuracy is 82.3% is derived by Algorithm 4.

From the experiments conducted and the analysis on them can draw conclusions. Algorithms 4 and 5, based on the proposed modifications of the k nearest neighbors, named Mk-NN modification reachs a higher value (reliability) of the score coefficient, i.e. higher matching rates between predicted classes and actual observation classes. In order to determine the advantages of both algorithms 4 and 5, it is sufficient to choose the variable shuffle = False, which means that the execution of the command kf = KFold (len (X), n_folds = 6, shuffle = False) is determined. In each of its execution, the division of n_folds equal parts of the set X is the same for both algorithms 4 and 5. Let us under these conditions perform an additional experiment on the same data sets and the results can be seen in Table 3. From the results in Table 3 shows that the results of the two algorithms coincide, i.e. it does not matter if we first determine the number of neighbors or first divide the main set into a training or test.

Table 3: Results from Algorithms 4 and 5 are the same when Shuffle = False

	Algorithm 4 k, score	Algorithm 5 k, score
Diabetes	k=19, 0.898	k=19, 0.898
German	k=17, 0.766	k=17, 0.766
Liver	k=9, 0.759	k=9, 0.759
Vehicle Silhouettes	k=7, 0.723	k=7, 0.723
winequality-red	k=15, 0.596	k=15, 0.596
winequality-white	k=17, 0.483	k=17, 0.483

5. Conclusion

In this paper we compare different algorithms for applying k-NN method which strive to achieve better efficiency in classifying test observations. The methodology proofs that the algorithms achieve better values of the controlled score parameters. The pair of train/test datasets that provides the highest score in terms of correctly predicted test observations against all test observations is selected for the final modelling of the dataset in question.

Experiments were carried out to compare the algorithms of the k nearest neighbor algorithm to classify big data sets. Clearly, algorithms 4 and 5 stand out for greater reliability of built-in models.

References

[1] Abdi, F, Khalili-Damghani, K & Abolmakarem, S (2018), Solving customer insurance coverage sales plan problem using a multi-stage data mining approach. *Kybernetes*, 47(1), 2-19. https://doi.org/10.1108/K-07-2017-0244.

[2] Andrade A, Silva JS, Santo J & Belo-Soares P (2012), Classifier approaches for liver steatosis using ultrasound images, *Procedia Technology* 5, 763-770. https://doi.org/10.1016/j.protcy.2012.09.084.

[3] Bagui, S, Bagui, S, Pal K. & Pal N (2003), Breast cancer detection using rank nearest neighbor classification rules, *Pattern Recognition*, 36(1), 25-34. https://doi.org/10.1016/S0031-3203(02)00044-4.

[4] Deng Z, Zhu X, Cheng D, Zong, M & Zhang S (2018), Efficient k NN classification algorithm for big data. *Neurocomputing*, 195, 143-148. https://doi.org/10.1016/j.neucom.2015.08.112.

[5] Gera C. & Joshi K (2015), A Survey on Data Mining Techniques in the Medicative Field. *International Journal of Computer Applications*, 113(13), 32-35. https://doi.org/10.5120/19888-1926.

[6] Halder A, Dey S & Kumar A (2015) Active Learning Using Fuzzy k-NN for Cancer Classification from Microarray Gene Expression Data. *Lecture Notes in Electrical Engineering*, 103-113. https://doi.org/10.1007/978-81-322-2464-8_8.

[7] James, G., Witten, D., Hastie, T. & Tibshirani, R (2013), An Introduction to Statistical Learning with Applications in R, *Springer*, http://www-bcf.usc.edu/~gareth/ISL/index.html (Access 2019). https://doi.org/10.1007/978-1-4614-7138-7_1.

[8] Jena M, Mishra SP & Mishra D, (2018), A survey on applications of machine learning techniques for medical image segmentation, *International Journal of Engineering & Technology*, 7(4), 4489-4495.

[9] Kulkarni SG & Babu MV (2013), Introspection of various K-Nearest Neighbor Techniques. *UACEE International Journal of Advances in Computer Science and Its Applications*, 3, 103-106.

[10] Kumar A (2016), Learning Predictive Analytics with Python. Packt Publishing, https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781783983261 (April 2019)

[11] Lin W, Ke, S & Tsai C, (2017), Top 10 Data Mining Techniques in Business Applications: A Brief Survey. *Kybernetes*, 46(7), 1158-1170, https://doi.org/10.1108/K-10-2016-0302.

[12] Ivanov I & Tanov V, (2018), Big Data Analytics Algorithms and Applications. Machine Learnings, ISBN 978-619-239-010-5, Sofia (in Bulgarian).

[13] Phogat M, Kumar D, (2018), A survey of machine learning techniques for genomic diseases and data sets, *International Journal of Engineering & Technology*, 7(4), 5533-5538.

[14] Shamitha SK, Ilango V, (2018), A survey on machine learning techniques for fraud detection in healthcare, *International Journal of Engineering & Technology*, 7(4), 5862-5868.
[15] Shazmeen SF, Baig MMA & Pawar MR (2013), Performance Evaluation of Different Data Mining Classification Algorithm and Predictive Analysis, *IOSR Journal of Computer Engineering*, 10(6), 1-6. https://doi.org/10.9790/0661-1060106.
[16] The Python Sklearn Library, 2019, http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (Access 2019)
[17] Umasankar P, Thiagarasu, V, (2018), Proposing a new methodology on vague association rule mining for the diagnosis of heart disease hesitation patterns, *International Journal of Engineering & Technology*, 7(4), 5851-5855.
[18] Zhou Y, Li Y & Xia S (2009) An Improved KNN Text Classification Algorithm Based on Clustering. *Journal of Computers*, 4(3), 230-237. https://doi.org/10.4304/jcp.4.3.230-237.