RJNMSR
Review Journal of Neurological & Medical Sciences Review

Volume 3, Issue 1, 2025

PREDICTION OF BREAST CANCER USING K-NEAREST NEIGHBOR CLASSIFIER WITH OPTIMAL *K* SELECTION

Muhammad Hamraz[*1], Nosheen Faiz[2], Naz Gul[3], Soofia Iftikhar[4]

[*1,2,3]*Department of Statistics, Abdul Wali Khan University, Mardan, 23200, Pakistan*
[4]*Department of Statistics, Shaheed Benazir Bhutto Women University, Peshawar, Pakistan*

[*1]mhamraz@awkum.edu.pk

Corresponding Author: *
Muhammad Hamraz
DOI: https://doi.org/10.5281/zenodo.18323112

Received	Accepted	Published
28 January 2025	12 March 2025	26 March 2025

ABSTRACT

In many scientific fields, machine learning methods have been widely used, but their use in medical literature is limited, partly due to technical difficulties. This research focuses on a machine learning method for predicting breast cancer with the aim of finding the optimum value of k in the k-Nearest neighbor classifier while dividing the data into different training and testing parts. The dataset used in this research work is "nki70" taken from the open.ml (https://www.openml.org/d/1147) machine learning repository having 77 features and 144 observations. The data set is a binary class problem with 48 observations of class 1 that represent breast cancer, and 96 observations of the other class 0, representing no breast cancer patients. The method considered in this research is the k-nearest neighbor algorithm. For selecting the best value of k, the performance metric, i.e., classification error rate, has been used. Results are given in the form of an average of 500 runs of the experiments on the different random training and testing sets. Furthermore, box plots of the results are also constructed. From the results of the analysis, it has been observed that the best value of k for which the k-NN classifier produced minimum error is k=1.

Keywords: *k-nearest neighbor, breast cancer, optimum value, classification, binary class problem.*

1. Introduction

Cancer refers to any of a large number of diseases described by the production of uncontrollably dividing abnormal cells that are capable of damaging and destroying healthy body tissue. Cancer also has the potential to spread all over the body. There are several cancer forms [1]. Cancer is not a single disease; it can originate in the lungs, breast, colon, or even in the blood [2]). Although cancer share certain common characteristics, they differ in the ways they grow, spread, and reproduce [3]. Breast cancer is a disease in which malignant cells grow uncontrollably in the tissues of the breast [4]. It is the most prevalent cancer among women in the United States, excluding skin cancer [5]. Although breast cancer occurs in both males and females, it is significantly more common in females [6]. Increased awareness, along with strong research foundations, has contributed to improvements in the diagnosis of breast cancer [7]. While breast cancer remains a major health concern, mortality rates have steadily declined due to early detection, advances in targeted therapies, and improved treatment strategies [8,5].

The breast consists of three main components: lobules, ducts, and connective tissue [4]. Lobules are the glands responsible for producing milk, and many breast cancers originate in the ducts or

lobules [9]. Breast cancer can spread beyond the breast through blood vessels and lymphatic channels, and when the cancer spreads to distant parts of the body, it is referred to as metastasis [10].

Recent years have witnessed a lot of work on machine learning models to accurately classify the tissue samples related to breast cancer into their true classes. These models include random forest, *k*-NN, and Support vector machine, etc.

Earlier experience of machine learning in breast cancer was centred on classification and diagnosis, especially on the Wisconsin Breast Cancer Dataset. Conventional algorithms like k-Nearest Neighbors (k-NN), Decision Trees, and Naive Bayes were shown to be promising in the differentiation of benign and malignant tumors [11]. One of them is k-NN, which is highly researched because of its simplicity and ability to perform pattern recognition tasks, but has a high cost of computation with large data sets [12].

Support Vector Machines (SVMs) became a potent alternative due to their high ability to generalize and perform effectively in high-dimensional space. A number of studies noted better diagnostic performance of SVMs in comparison to the classical statistical techniques, particularly when they are used together with kernel functions [13]. On the same note, ANNs have been widely used in the diagnosis and prognosis of breast cancer, and have proven to be highly accurate, due to their ability to model nonlinear relationships among clinical and gene expression characteristics [14]. As high-throughput genomic technologies emerged, research in breast cancer shifted to study the expression of genes, and machine learning models have become vital in the discovery of biomarkers and classification of subtypes. The collection of feature selection algorithms with ML classifiers has been demonstrated to enhance the accuracy of prediction as well as dimensionality and complexities are minimized [15]. Particularly, random Forest (RF) has been popular with its resistance to noise, capacity to deal with thousands of variables, and feature importance measures [16].

Over the last few years, the field of deep learning (DL) has experienced further development of breast cancer research, particularly in medical imaging. Convolutional Neural Networks (CNNs) have demonstrated the state of art in mammography, ultrasound, and histopathological image processing by automatically discovering discriminative features [17]. It has been demonstrated that traditional ML methods are less effective at image-based tasks of breast cancer detection than CNN-based models [18]. Additionally, ensemble and hybrid methods that involve using two or more classifiers or the combination of clinical, imaging, and genomic data have been suggested in order to improve predictive performance. This is because such models are less biased, less variable, and more robust and, thus, should be used with complex and heterogeneous breast cancer data sets [19]. Although progress has been made, there are still problems such as the imbalance in the classes, the explainability of the multi-faceted models, and the potential to generalize the results across societies. As a consequence, recent investigations focus on explainable AI, selection of features optimally, and hybrid ML models to overcome these limitations and enhance clinical applicability [20].

The current study aims to use a *k*-NN classification model to study the "nki70" data set and to identify the optimum value of *k* that produces the minimum classification error rate. Moreover, this study also investigates the effect of training set size on the efficacy of the *k*-NN model.

1.1 *k*- nearest neighbor classifier

The K-Nearest Neighbors (k-NN) algorithm is an easy and effective non-parametric and instance-based algorithm of machine learning that is extensively employed in classification and regression. It is based on the assumption that similar data points are likely to be near each other in a feature space. Given an unobserved observation, k-NN finds the *k* nearest training examples in terms of a distance, i.e., most often Euclidean distance, and classifies the unobserved instance by the

majority of the closest neighbors. The ease of implementation and the fact that k-NN can effectively solve pattern recognition problems, especially in cases where boundaries to the decision are not regular, are some of the key benefits of the algorithm. It has worked in medical diagnosis, including breast cancer, where it is capable of accurately categorizing tumors as benign or malignant using clinical or imaging characteristics. But the performance of k-NN is very sensitive to the number of k, the distance measure, and the feature scale, and the cost of computation of k-NN is high with a large dataset, as all the training samples need to be stored and searched during prediction.

1.2 How does the *k*-NN algorithm work?

The k-NN classifier identifies the class of a test sample point by using the following procedure

1) Determine parameter k = number of nearest neighbors.
2) Find the distance between the test points and all training samples.
3) Based on the minimum distance of k, sort the distances and decide the nearest neighbors.
4) Select the nearest neighbor's category.
5) Use the simple majority of the class of nearest neighbors as the prediction value of the query instance.

The commonly used metric for finding the distance between the test query and training is Euclidean distance. For any two points $x = (x1, x2, \dots, xn)$ and $y = (y1, y2, \dots, yn)$, the Euclidean is given as.

$$d(x, y) = \sqrt{(x - y)^2} \qquad (1)$$

2. Methodology

A detailed description of the research methodology employed in this research is provided in this section. The research methodology is divided into two parts: (i) choosing the best value of k in the *k*-NN classifier that will minimize the classification error rate. (ii) Assessment of the classification error rate of the *k*-NN classifier for different values of k in combination with training set size.

2.1 Micro Array Data

The development of microarray technology provides easy monitoring of thousands of gene expressions simultaneously, and the gene expression data obtained from this technology is valuable for cancer classification. Microarray technology is also used in several studies for the diagnosis of breast cancer. For thousands of genes, but with a limited number of observations, microarray technology produces large data sets with expression values. In general, these data are arranged in a matrix of n rows and m columns. Whereas the columns represented genes, the rows represented samples. This $n \times m$ gene profile cannot be analyzed manually because of its dynamic existence. Such complicated data sets can't be analyzed manually; however, computer software is used to analyze these huge data sets. However, innovative statistical techniques and advanced computing software are essential for the successful analysis of microarray data. Current bioinformatics tools and their promising applications play a crucial role in analyzing data from microarray experiments.

2.2 Data description

The breast cancer dataset has been taken from the openML repository, or it can be downloaded from R software. The breast cancer dataset has a total of 144 observations and 77 features. Moreover, the data can be accessed via (https://www.openml.org/d/1147).

2.3 Software and packages

In this research, R software is used for the analysis, which can be downloaded from https://www.r-project.org/. The package used in this study is **"class"**. The full title of the class package is 'Functions for Classification'. It provides various functions for classification, including k-nearest neighbour, Learning Vector Quantization, and Self-Organizing Maps. The authors of the package are Brian Ripley [aut, cre, cph] and William Venables [cph].

2.4 Classification method

In this work, a single classifier is used for evaluation, namely the k-nearest neighbor (*k*-NN) classifier.

2.5 *k*-Nearest Neighbor (*k*-NN)

The k-nearest neighbor (k-NN) algorithm is a simple method that stores all available cases and classifies new instances based on a similar measure, such as distance functions. As a non-parametric method, k-NN has been widely used in statistical estimation and pattern classification since the early 1970s. The k-NN algorithm can also be applied to regression problems. It is an instance-based approach that focuses on the data points most similar to a given query point to perform classification. To provide an 'educated guess' for an unclassified instance, the algorithm uses labeled training data. The main features of the k-NN algorithm are as follows:

i. *k*-NN is a Supervised Learning algorithm that uses a labeled input data set to predict the output of the data points.
ii. It is one of the best algorithms for machine learning and can be easily applied for a complex range of problems.
iii. It is largely focused on the similarity of functions. K-NN verifies how similar a data point is to its neighbors and categorizes the data point into the class to which it is most similar.

2.6 Error Estimation Methods and Evaluation

To determine the efficiency of the classifier on both i.e. training set size and different values of *k*, the classification error rate has been used, which is given below.

$$Error\ Rate = \frac{FP+FN}{TP+TN+FN+FN}, \quad (2)$$

where $FP, FN, TP\ and\ TN$ represent false positive, false negative, true positive and true negative cases respectively.

3. Results and discussion

This section presents the results based on the analysis done in this work. The classification error rates of classification for different values of *k* are calculated. For further assessment, the Box plots of the results are also constructed. Table 1 shows the error rate of classification of the *k*-NN with a 90% training and 10% testing set size. It shows that the value of *k*=1 gives the best results, because the classification error rate is low, i.e., 0.0549. When *k* = 3, the error classification rate is 0.1624; for *k*=4 the classification error rate is 0.1713. Similarly, for *k*=5 the classification error rate is 0.1551, while for *k*=6 the classification error rate is 0.1597. It is obvious from the above results that as we increase the value of *k*, the error rate of classification increases. Hence, in this case, the finest value of *k* is 1, i.e, k=1. Likewise, for an 80% training and 20% testing set size, the optimal *k* value is "*k*=1". Which has a minimum classification error as compared to other values of *k*. When *k*=3, the error rate is 0.1620, for *k*=4 the error rate is 0.1737, for *k*=5 the error classification rate is 0.1551. Similarly, for *k*=6 the classification error rate is 0.1643, while for *k*=7 the classification error rate is 0.1540. It is evident from the above results that as we increase the value of *k*, the error rate of classification generally increases; therefore, the optimum value of *k* in this case is *k*=1. Table 3 shows the classification error rates of k-NN with 70% training and 30% testing data set size. It shows that *k*=1 and *k*=4 give the same results, which is used in the *k*-NN model, because the classification error rate is low, i.e., 0.0845. When *k* = 3, the error rate is 0.1350; for *k*=5 the error rate is 0.1580. Similarly, for *k*=6, the error rate is 0.1633, while for *k*=7, the classification error rate is 0.1610. It is obvious from the above results that as we increase the value of *k*, the error rate generally increasing, hence in this case the significant values of k are 1 and 4, i.e, k=4 and k=1, both are good. Table 4 shows the classification error rates of *k*-NN with 60% training and 40% testing data set size. It shows that *k*=1 gives the best results, which is used in the *k*-NN model, because the classification error rate is low, i.e., 0.1139. When *k* = 3, the error rate is 0.1742; for *k*=4 the error classification rate is 0.1829. Similarly, for *k*=5 the error rate is 0.1590, while for *k*=6 the classification error rate is 0.1705. It is clear from the above results that as we increase the value of *k*, the classification error rate generally increases; therefore, the best value of *k* in this case is *k*=1. Table 5 shows the classification error rates of *k*-NN with 50% training and 50% testing set sizes. It shows that *k*=1 gives the best results, which is used in the *k*-NN model, because the classification error rate is low, i.e., 0.1340. When *k* = 3, the error rate is 0.1794; for *k*=4 the error rate is 0.1886. Similarly,

for k=5, the error rate is 0.1734, while for k=6, the classification error rate is 0.1830. It is clear from the above results that as we increase the value of k, the classification error rate generally increases; therefore, the best value of k in this case is k=1. Finally, we observed that overall, from the results given in Tables 4.1-4.5, the best value of k, in all the situations, is 1.

Table 1: Classification error rates when the data is *partitioned* into 90% training and 10% testing parts, for different values of *k*.

Values of k	Classification Error Rates
1	0.0549
3	0.1624
4	0.1713
5	0.1551
6	0.1597
7	0.1530
10	0.1672
11	0.1703
13	0.1635
14	0.180
15	0.1894
17	0.1915
18	0.1877
20	0.1897

Table 2: Classification error rates when the data is *partitioned* into 80% training and 20% testing parts, for different values of k.

Values of k	Classification Error Rates
1	0.0549
3	0.1620
4	0.1737
5	0.1551
6	0.1643
7	0.1540
10	0.1626
11	0.1715
13	0.1802
14	0.1904
15	0.1872
17	0.1871
18	0.1803
20	0.1828

Table 3: Classification error rates when the data is *partitioned* into 70% training and 30% testing parts, for different values of *k*.

Values of k	Classification Error Rates
1	0.0845
3	0.1350
4	0.0845
5	0.1580
6	0.1633
7	0.1610
10	0.1794
11	0.1808
13	0.1898
14	0.1923
15	0.1882
17	0.1819
18	0.1932
20	0.1963

Table 4: Classification error rates when the data is *partitioned* into 60% training and 40% testing parts, for different values of *k*.

Values of k	Classification Error Rates
1	0.1139
3	0.1742
4	0.1829
5	0.1590
6	0.1705
7	0.1630
10	0.1710
11	0.1893
13	0.1901
14	0.1990
15	0.1944
17	0.1987
18	0.2053
20	0.2033

Table 5: Classification error rates when the data is *partitioned* into 50% training and 50% testing parts, for different values of *k*.

Values of k	Classification Error Rates
1	0.1340
3	0.1794
4	0.1886
5	0.1734
6	0.1830
7	0.1839
10	0.1921
11	0.1936

13	0.1989
14	0.2025
15	0.2035
17	0.2084
18	0.2098
20	0.2139

For further assessment, box plots of the results are also constructed. The box plots are shown in Figures 1–5. From Figure 1, it is evident that for k = 1, the average error of the k-NN classifier is less than the average errors for other values of k. When k = 3, the average error is 0.08; for k = 4, the average error is the same as the average error for k = 3. For k = 5, the average error has decreased, whereas for k = 6, the average error has increased. For k = 7, the average error has decreased as compared to k = 6. Further, taking the value of k = 10, it has been observed that there is a slight increase in the average error. Furthermore, for k = 11, the average error is 0.19; for k = 13 and k = 14, the average errors are the same as the average error for k = 11. For k = 15, the average error has increased, while for k = 17, k = 18, and k = 20, the average errors remain the same, as there is no increase or decrease in the error values with increasing k.

In general, when the data are divided into 90% training and 10% testing parts, the best value of k in this situation is k = 1. Figure 2 represents the average error rates provided by the k-NN classifier for different values of k. When k = 1, it provides the minimum error rate as compared to other values of k, i.e., 0.02. While for k = 3, the variation in the results is larger as compared to k = 4, the average errors are the same.

For k = 5, the average error has decreased; for k = 6, the average error has increased; for k = 7, the average error has decreased compared to k = 6. For k = 10, the average error has increased. Furthermore, for k = 11, the average error is 0.15, and similarly, for k = 13, k = 14, k = 15, k = 17, k = 18, and k = 20, the k-NN model provides the same results, as there is no fluctuation in the error values with increasing k. Finally, for k = 1, the variation in the cross-validation results is minimal compared to other values. In general, when the data is divided into 80% training and 20% testing, the best value of k is k = 1.

In Figure 3, it is clear that for k = 1, the average error of the k-NN classifier is lower than the errors for other values of k. When k = 3, the average error is 0.14, while for k = 4, the average error has also increased. For k = 5, the average error has decreased; if k = 6 or k = 7 is selected, the average error is similar to that of k = 5. For k = 10, there is a slight increase in the average error. Moreover, for k = 11, the variation in the results is smaller compared to k = 13, although the average errors are the same. For k = 14, k = 15, k = 17, and k = 18, the average errors are the same as that of k = 13. For k = 20, the average error increases. Thus, when the data is divided into 70% training and 30% testing, the best value of k is k = 1.

In Figure 4, it is evident that for k = 1, the average error of the k-NN classifier is lower than for other values of k. When k = 3, the average error is 0.18; for k = 4, the average error is the same as for k = 3. If k = 5, 6, 7, 8, or 10 is selected, the average errors decrease and increase step by step. For k = 11, the average error is 0.19; for k = 13, the average error is the same as for k = 11. For k = 14, the average error increases; for k = 15 and k = 17, the average errors are approximately the same. For k = 18, the average error increases, and for k = 20, the average error is the same as for k = 18. Therefore, when the data is divided into 60% training and 40% testing, the best value of k is k = 1.

From Figure 5, it is evident that for k = 1, the average error of the k-NN classifier is lower than for other values of k. When k = 3, the average error is 0.19; for k = 4, the average error is the same as for k = 3. For k = 5, the average error decreases; for k = 6 and k = 7, the average errors are approximately the same. For k = 10, there is a slight increase in the average error. For k = 11, the average error is 0.20; for k = 13, the average error remains the same. For k = 14, the average error increases compared to k = 13; for k = 15, 17, 18, and 20, the average errors are the same, with no fluctuation as k increases. Thus,

when the data is divided into 50% training and 50% testing, the best value of k is k = 1.

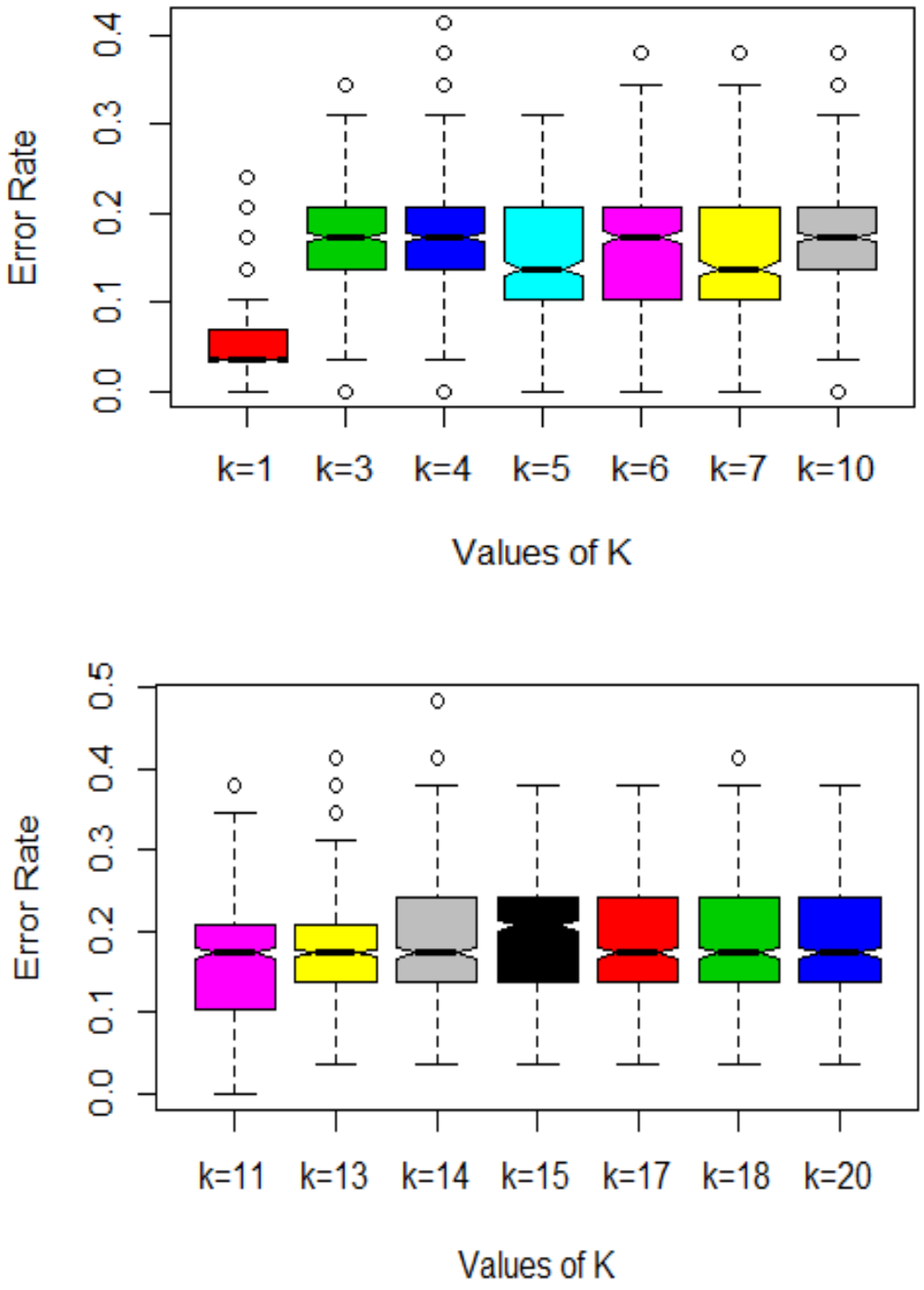

Figure 1: Box plots for different values of k, when the data is partitioned into 90% training and 10% testing parts

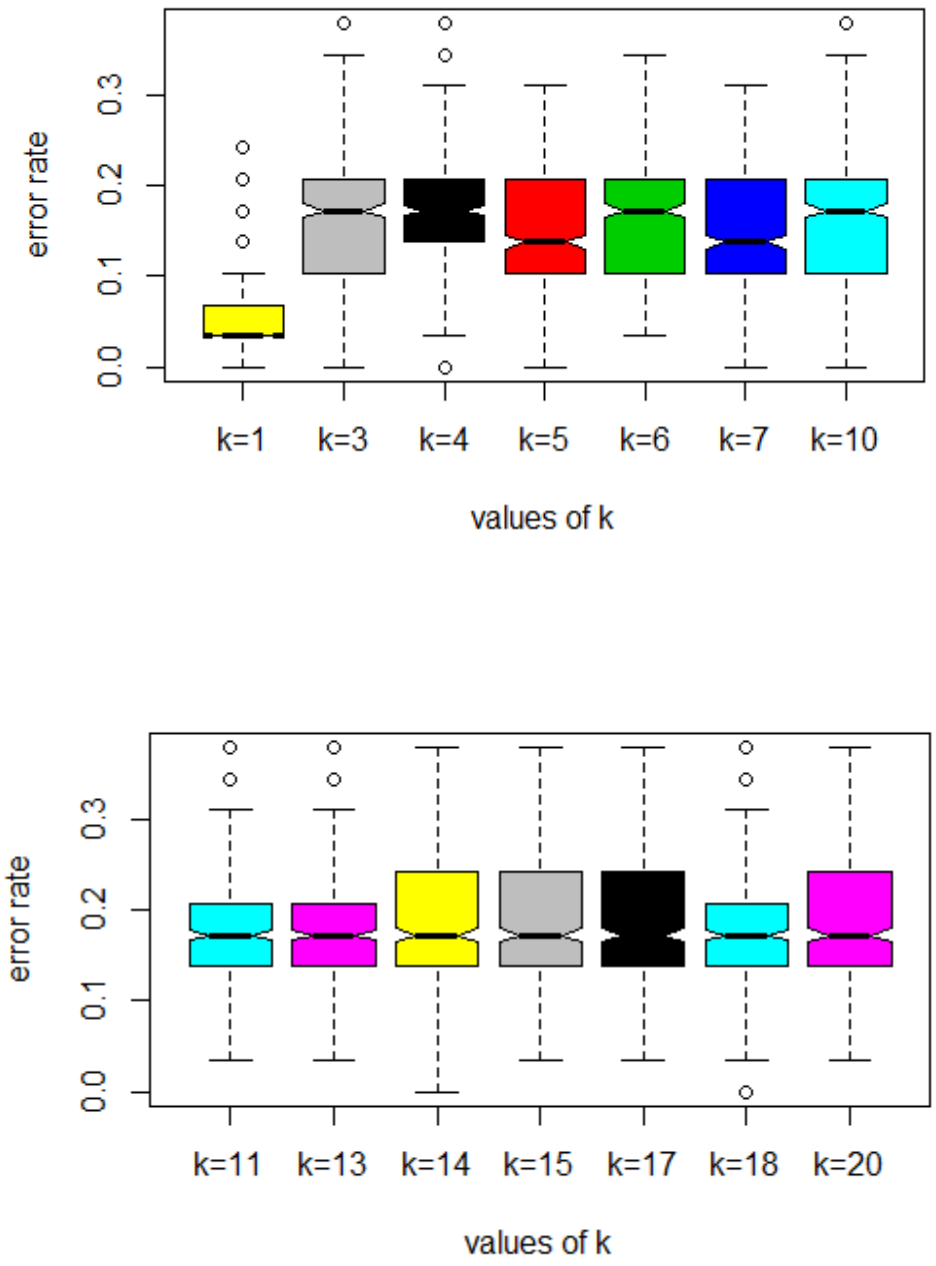

Figure 2: Box plots for different values of *k*, when data is partitioned into 80% training and 20% testing parts.

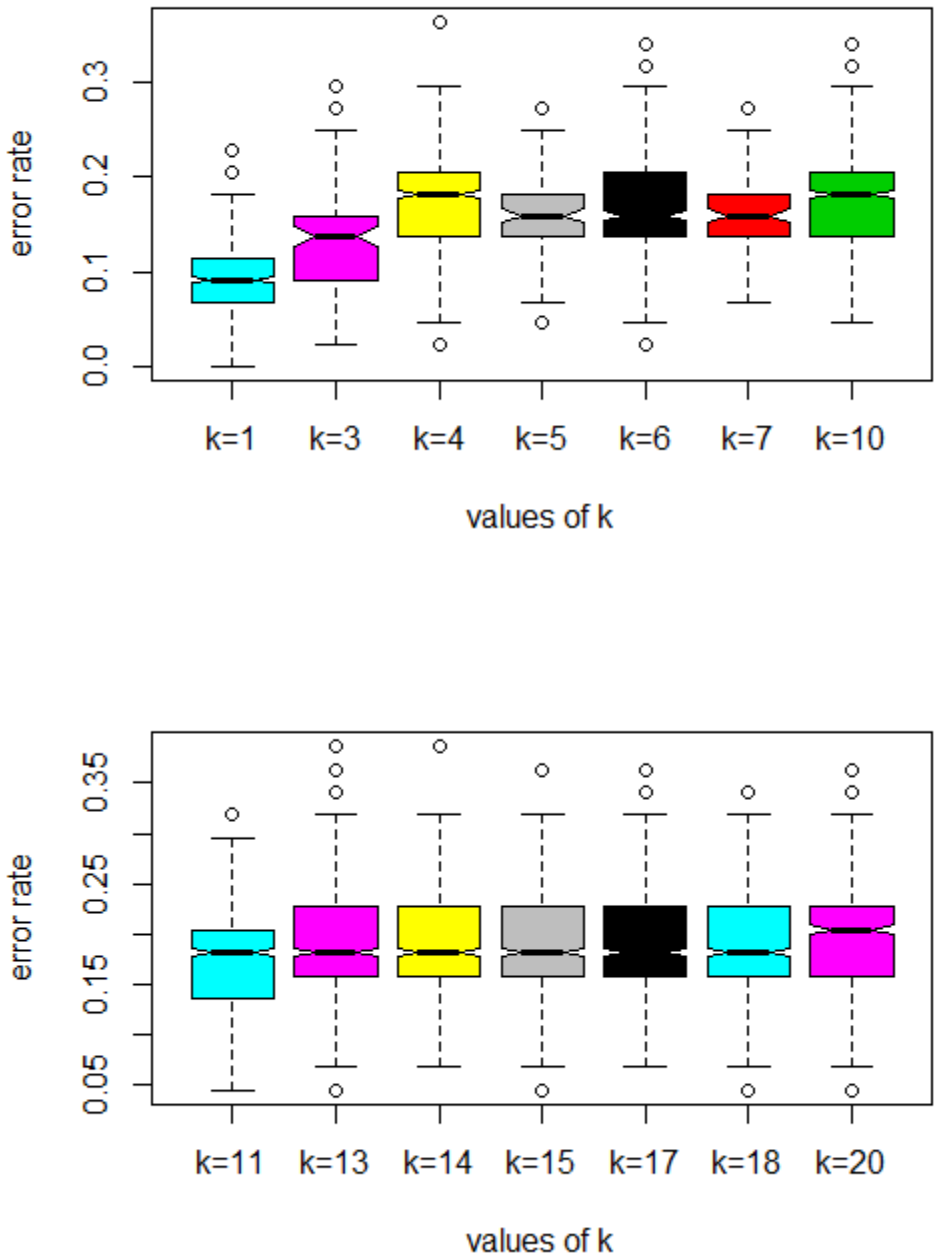

Figure 3: Box plots for different values of *k*, when data is partitioned into 70% training and 30% testing parts

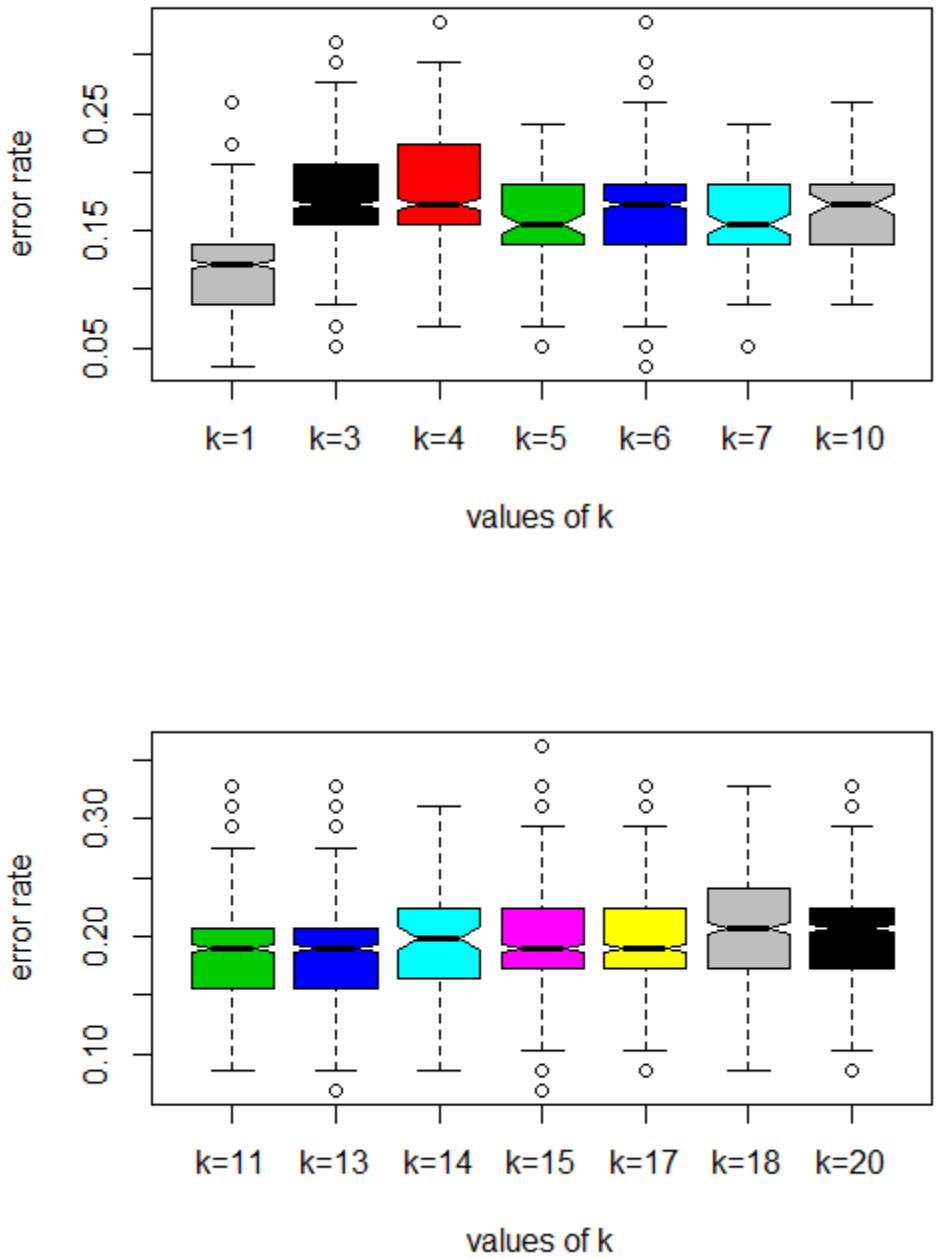

Figure 4: Box plots for different values of *k*, when data is partitioned into 60% training and 40% testing parts.

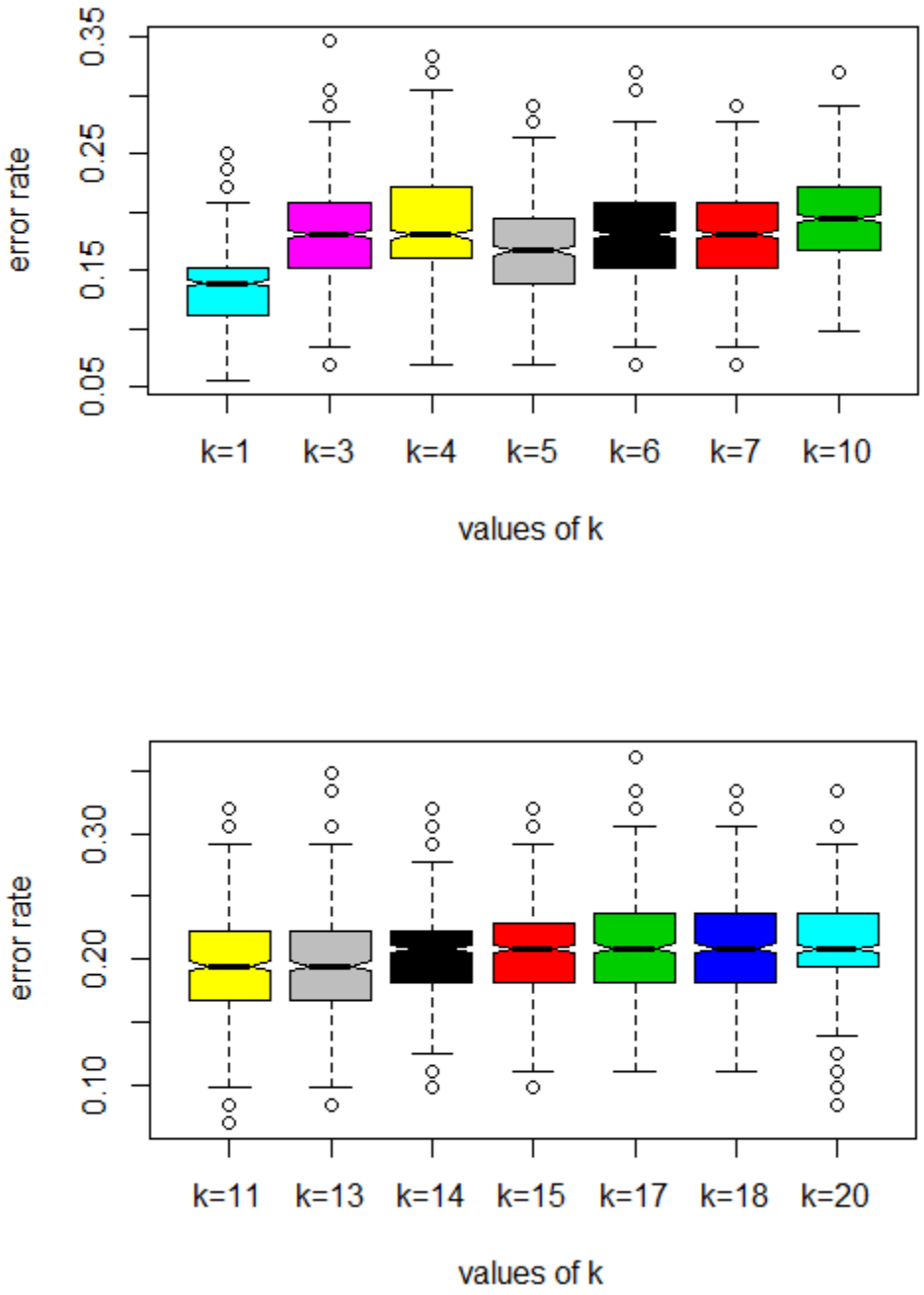

Figure 5: Box plots for different values of *k*, when data is partitioned into 50% training and 50% testing parts.

4. Conclusion

After analyzing the breast cancer data using the k-NN classifier, we have observed that different values of k have a significant effect on the classifier's accuracy. It has also been observed that the training set size, along with the value of k, has a great impact

on the classification error rate of the k-NN classifier. Initially, the data was divided into two parts: the first part as the training set and the second part as the testing set.

For assessing the best value of k, the performance metric used is the classification error rate. Different training set sizes have been discussed in this research work. For example, when the data was divided into 90% training and 10% testing, the best value of k was 1, because the classification error for k = 1 is 0.0549. Similarly, when the data was divided into 80% training and 20% testing, the best value of k was again 1, because the classification error for k = 1 is also 0.0549. In this case, the training and testing set size had no effect, as the classification error rates are the same.

Furthermore, when the data was divided into 70% training and 30% testing, the best values of k were 1 and 4, because the classification error rates for k = 1 and k = 4 are both 0.0845. Moreover, when the data was divided into 60% training and 40% testing, the best value of k observed was 1, with a classification error of 0.1139. Finally, when the data was divided into 50% training and 50% testing, the best value of k observed was 1, with a classification error rate of 0.1340.

Furthermore, for different values of k, box plots of the results were also constructed. When the data was divided into 90% training and 10% testing, it was observed that the best value of k was 1, with an average classification error of 0.02 (Figure 1). Similarly, for 80% training and 20% testing, the best value of k was 1, with an average classification error of 0.02 (Figure 2). When the data was divided into 70% training and 30% testing, the best value of k was 1, with an average classification error rate of 0.09 (Figure 3). For 60% training and 40% testing, the best value of k was 1, with an average classification error rate of 0.13 (Figure 4). Finally, when the data was divided into 50% training and 50% testing, the best value of k was 1, with an average classification error rate of 0.14 (Figure 5).

Overall, from the results presented in Tables 1-5, it is clear that the best value of k in all situations is 1 for the breast cancer data, which is also supported by the constructed box plots.

REFERENCES

National Cancer Institute. (2023). Cancer. Retrieved from National Cancer Institute Dictionary of Cancer Terms.

American Cancer Society. *Breast Cancer Facts & Figures*. 2024.

Berry, D.A., et al. (2005). Effect of screening and adjuvant therapy on mortality from breast cancer. *New England Journal of Medicine*, 353, 1784–1792.

Chaffer, C.L., & Weinberg, R.A. (2011). A perspective on cancer cell metastasis. *Science*, 331, 1559–1564.

DeSantis, C.E., et al. (2019). Breast cancer statistics. *CA: A Cancer Journal for Clinicians*, 69, 438–451.

Hanahan, D., & Weinberg, R.A. (2011). Hallmarks of cancer: the next generation. *Cell*, 144, 646–674.

Harbeck, N., et al. (2019). Breast cancer. *Nature Reviews Disease Primers*, 5, 66.

Siegel, R.L., et al. (2024). Cancer statistics, 2024. *CA: A Cancer Journal for Clinicians*, 74, 17–48.

World Health Organization (WHO). *Breast cancer fact sheet*. 2023.

Weinberg, R.A. (2014). *The Biology of Cancer*. 2nd ed., Garland Science.

Wolberg, W. H., Street, W. N., & Mangasarian, O. L. (1994). Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters, 77(2–3), 163–171.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27.

Vapnik, V. (1998). Statistical Learning Theory. Wiley.

Lisboa, P. J. G., & Taktak, A. F. G. (2006). The use of artificial neural networks in decision support in cancer. European Journal of Cancer, 42(10), 1494–1500.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157–1182.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Litjens, G., et al. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88.

Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016). Breast cancer histopathological image classification using convolutional neural networks. International Joint Conference on Neural Networks (IJCNN), 2560–2567.

Polat, K., & Güneş, S. (2007). Breast cancer diagnosis using least square support vector machine. Digital Signal Processing, 17(4), 694–701.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD Conference, 1135–1144.