kNN Classifier

Gian Marco Balia
Robotic Engineering - University of Genoa
s4398275@studenti.unige.it

***Abstract*—The k-Nearest Neighbors (kNN) classifier is a supervised machine learning algorithm known for its simplicity and versatility, particularly effective in classification tasks. For instance, it is widely used in the medical field which highlights the importance of ontological approaches to optimize feature selection and enhance the robustness of kNN in complex contexts.**

This report presents the implementation of kNN, emphasizing data preprocessing, relevant feature selection, and the optimization of the k parameter. The algorithm was applied to a reference dataset to evaluate its performance in terms of accuracy and precision. The results demonstrate how the choice of k significantly influences the trade-off between predictive accuracy and computational complexity.

***Index Terms*—kNN Classifier, accuracy**

I. Introduction

In the rapidly evolving field of machine learning, the k-Nearest Neighbors (kNN) algorithm has emerged as a fundamental yet powerful tool for classification and regression tasks. Its intuitive approach—relying on proximity within the feature space to make predictions—makes kNN particularly attractive for applications where interpretability and simplicity are key. Despite its straightforward nature, the algorithm's performance is deeply influenced by factors such as data preprocessing, feature selection, and the choice of the k parameter, which determines the number of neighbors considered during classification.

One of the primary strengths of kNN lies in its versatility. It has been successfully applied in different fields, as image recognition, fraud detection, and medical diagnostics. A study *"Fuzzy Decision Ontology for Melanoma Diagnosis Using kNN Classifier"* [1] underscore the algorithm's role in improving diagnostic accuracy through tailored feature selection strategies. However, kNN is not without limitations. It is sensitive to noise, computationally intensive for large datasets, and may struggle with imbalanced data.

There were implemented and evaluated the kNN algorithm focusing on its practical aspects, performance optimization, the impact of preprocessing techniques, the role of feature selection in reducing dimensionality, and the trade-offs involved in determining the optimal number of neighbors k.

By providing a comprehensive analysis of kNN's strengths, limitations, and potential improvements, this report aims to offer valuable insights for leveraging the algorithm effectively in real-world applications.

II. Material and methods

A. Data processing

This study is based on the analysis of a dataset describing various types of wines through their chemical characteristics and corresponding classifications. Specifically, the dataset is divided in two subsets: *wine.data*, which contains the chemical measurements associated with each wine sample (i.e., *Alcohol*, *Malic acid*, *Ash*, *Alcalinity of ash*, *Magnesium*, *Total phenols*, *Flavanoids*, *Nonflavanoid phenols*, *Proanthocyanins*, *Color intensity*, *Hue*, *OD280/OD315 of diluted wines*, *Proline*), and $wine.target$, which identifies the specific class to which each wine in the analysis belongs. Subsequently, both subsets were split and randomized to ensure accurate analysis. Specifically, the following subsets were generated:

- x_{train}, containing 80% of the data;
- x_{test}, containing 20% of the data;
- y_{train}, containing 80% of the targets;
- y_{test}, containing 20% of the targets.

In addition, prior to the analysis, it was necessary to normalize the obtained subsets. This step is crucial because the analysis involves calculating distances between different values. Given that the dataset contains variables with differing ranges of values, it is essential to scale the data to a uniform range to ensure that all features contribute equally to the computation of distances. These subsets were then used for training and testing the kNN Classifier implemented in our study.

B. kNN Classifier

The implementation of k-Nearest Neighbors (kNN) algorithm is based on computing of normalized linear distance between the points of the training set and the evaluated test's point. After computing the distance, the algorithm predicted the class to which points belong based on the number of neighbors k.

C. Model evaluation

The model was evaluated computing its accuracy in relation to the specific k parameter considered during the analysis and model training. As mentioned earlier, the k parameter represents the number of nearest neighbors taken into account for classifying a new data point, determining how many points in the training dataset are chosen to decide the class of a sample. The results obtained, specifically the model's accuracy for each value of k, are displayed in Figure 1. Another parameters used to evaluate the model are: the *error rate*, which represent the percentage of incorrect predictions; the *precision* that characterize the proportion of true positives (correctly classified instances) among all positive predictions made by the classifier; the *recall* which is the proportion of true positives among all actual positive instances; the *F1 Score* that represent the harmonic mean of precision and recall, providing a balanced measure of both. The last three parameters are obtained with a *confusion matrix*. A *confusion matrix* is a table used to evaluate the performance of a classification model by comparing its predictions against the actual outcomes. It provides a detailed breakdown of prediction accuracy across different classes, offering insight into the model's strengths and areas for improvement. The confusion matrix is structured as follows:

$$C = \begin{pmatrix} \text{TP} & \text{FP} \\ \text{FN} & \text{TN} \end{pmatrix}$$

where the *true positives* (TP) are the cases where the model correctly predicted the positive class; the *true negatives* (TN) are the cases where the model correctly predicted the negative class; the *false positives* (FP) are the cases where the model incorrectly predicted the positive class (Type I error); and the *false negatives* (FN) is the cases where the model incorrectly predicted the negative class (Type II error).

TABLE I
Class 1 Statistics over all k values

Metric	Mean	Median	AAD	1st Quantile	3rd Quantile
Precision	0.987879	1	0.0198347	1	1
Recall	0.991736	1	0.0135237	1	1
F1 Score	0.989797	1	0.0129861	0.971941	1

TABLE II
Class 2 Statistics over all k values

Metric	Mean	Median	AAD	1st Quantile	3rd Quantile
Precision	0.96558	0.978261	0.0344203	0.916667	1
Recall	0.941558	0.964286	0.0584416	0.8571431	1
F1 Score	0.953401	0.943218	0.0129612	0.943218	0.971223

TABLE III
CLASS 3 STATISTICS OVER ALL k VALUES

Metric	Mean	Median	AAD	1st Quantile	3rd Quantile
Precision	0.949495	1	0.0550964	0.888889	1
Recall	0.983766	1	0.0177096	0.964286	1
F1 Score	0.966295	0.962925	0.00612813	0.962925	0.962925

TABLE IV
TOTAL STATISTICS OVER ALL K VALUES FOR OF ALL CLASSES

Metric	Mean	Median	AAD	1st Quantile	3rd Quantile
Precision	0.967651	0.96558	0.0364505	0.957537	0.976729
Recall	0.972353	0.983766	0.029891	0.962662	0.987751
F1 Score	0.969831	0.966295	0.0106918	0.959848	0.978046

III. RESULTS AND CONCLUSION

The parameter k determines how many neighbors are considered to make the classification. The value of k has a significant impact on the model's performance. If k is too small, the algorithm can be sensitive to noise in the data and may lead to overfitting, where the model fits too closely to the training data. On the other hand, if k is too large, it can reduce the model's ability to capture local variations in the data, resulting in under-fitting, where the model fails to adequately represent the complexities of the dataset. As shown in Figure 1, choosing the optimal value of k is crucial to balance accuracy and computational complexity. Computing *error rate* for all classes and doing the mean between the ten possible value for each k, were been obtained the following results:

- Class 1 = 0.9899;
- Class 2 = 0.2677;
- Class 3 = 0.5859.

It is important to highlighting that the error rate in the first class is very high ($\approx$100%). This could be due to a low number of data points on which the model is trained, causing the algorithm to be unable to handle the data correctly. The last important parameters used to evaluate the k-Nearest Neighbors (kNN) algorithm are presented in the tables below. From the results shown in the final table, it is evident that the model's predicted precision is strong. In general, the results from the tables demonstrate good performance of the kNN model in terms of precision, recall, and F1 Score. The means are high for all metrics, and the first and third quantiles confirm that the extreme values (both low and high) are fairly stable. Overall, the evaluation of the model is positive, with a good balance between the different metrics for each class and for the entire dataset

REFERENCES

[1] W. Abbes, D. Sellami, S. Marc-Zwecker, and C. Zanni-Merk, "Fuzzy decision ontology for melanoma diagnosis using KNN classifier," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 25 517–25 538, Jul. 2021.

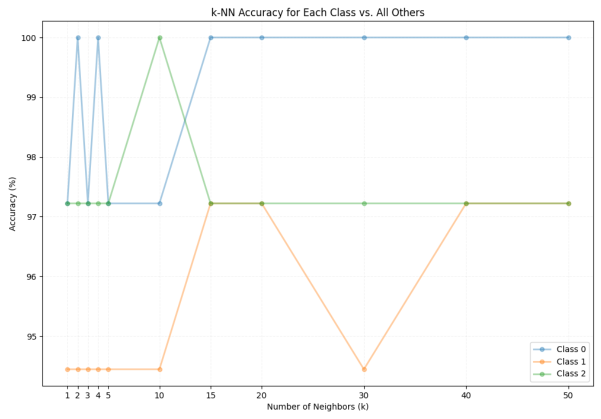

Fig. 1. Relationship between the number of neighbors k and the classification accuracy across all classes for the kNN classifier.