Impact of K-Nearest Neighbour on Classification Accuracy in KNN Algorithm Using Machine Learning

Abhishek Srivastava

Abstract Classification accuracy of the KNN algorithm is affected by the number of nearest neighbour for predicting points. The idea behind nearest neighbour classification consists in finding a number, i.e. the ‘*k*’—of training data point nearest in distance to a predicting data, which has to be labelled. The label of the new predicting data will be defined from these neighbours. Using a machine learning approach, the value of *k* will be fixed for model creation. In this paper, I analyse different numbers of neighbour so that classification accuracy can be maximized. Different distance metrics (Euclidean, Manhattan and Minkowski) can be used to measure the distance from the nearest data point.

Keywords Supervised machine learning · K-nearest neighbour

1 Introduction

Similarity measures play an important role while making classification using supervised machine learning. Similarity measures also create an important role while analysing some pattern of any class in the data set. KNN algorithms work on these patterns and check the new predicting data point match with which similarity based on its neighbour. Number of neighbours also plays an important role while classifying new data neighbour points. Majority of closely located neighbours help to classify new data points.

K-nearest neighbour uses different kind of distance metric to find distance between new data point with previously classified data point. These distance metrics are different in nature. Some of distance metric are used to calculate distance between coordinate points for example Euclidean some of distance metric uses distance been labelled data like yes and no like Hamming distance.

A. Srivastava (✉)
Department of Computer Science and Engineering, Sanskriti Institute of Management and Technology, Mathura, India
e-mail: mr.abhishk@gmail.com

R. Agrawal et al. (eds.), *Advances in Smart Communication and Imaging Systems*,
Lecture Notes in Electrical Engineering 721,
https://doi.org/10.1007/978-981-15-9938-5_35

Decision of number of nearest neighbour also plays an important role because by changing the nearest neighbour may invert the classification. With the help of supervised machine learning model can be train with the data set. With exploratory data analysis, we can also find some insight of data which can be further analysis.

2 Related Work

Thirunavukkarasu et al. [1] proposed a method based on KNN algorithm and finding its classification accuracy. He also found the training accuracy and testing accuracy. He also divided the data set into training and testing data in a ratio of 60% and 40%. In paper, misclassification rate is 0, while classification is shown with 100%. Kulkarni et al. [2] done image-based work on iris flower, he used grey level co-occurrence matrix to find region of interest, then classification is done based on KNN on fuzzy KNN algorithm. This is a new approach where images are used rather than textual data.

Tan [3] proposed a method for text categorization. He also proposed that each neighbour which has small class should have assigned a higher weight in comparison with each neighbour which has big class should have assigned a little weight. By using that approach, balancing is done between training samples, and this approach is named as neighbour weighted K-nearest neighbour (NWKNN).

Hastie and Tibshirani [4] proposed a K-nearest neighbour-based method which used discriminant adaptive nearest neighbour. Using training data set, he created a centroid and then find the decision line to classify the object after centroid. Formation object can be classified based on neighbourhood. Weinberger and Saul [5] proposed a new distance metric Mahalanobis distance for KNN classification. They also proposed that Euclidean distance does not play an important role in face recognition as for gender identification, that is why here Mahalanobis distance plays an important role in classification.

3 Proposed Model

KNN algorithm calculates distance between new data point and its neighbour of training data point. It arranges the distance in increasing order. Once the distances are arranged in increasing order, then predecided k-nearest neighbour can be found. Out of these K, the majority of votes are considered of different classes. The dominant class is labelled assigned to new data points.

Step 1: Select any value of nearest neighbour for classification.
Step 2: Calculate distance between the new point and its K neighbour.
Step 3: Sort the calculated distance.
Step 4: Count the data point with category wise among k neighbours.

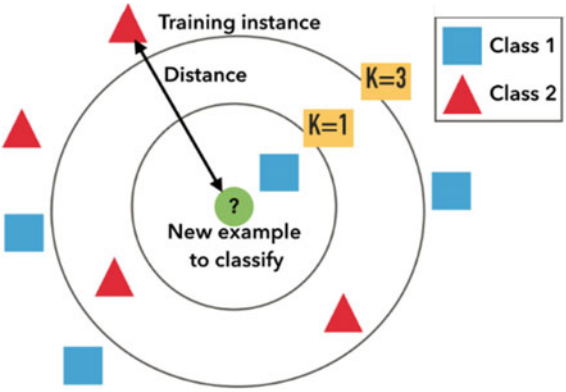

Fig. 1 Showing training data with different K values

Step 5: New data points will belong to the class that has the most neighbours (Fig. 1).

As shown in Figure, there are two classes blue square and red triangle, and we have to classify the green circle, lets first take $k = 1$ then green circle classify as blue circle while if we decide the $k = 3$ in that case one blue square and two red triangle. So, one conclusion we draw is that in the case of an odd number of neighbours there will never be a tie [1, 2].

4 Different Distance Metric Used in KNN

4.1 Minkowski Distance

Minkowski distance is used to calculate the distance between two points. Minkowski formula can be converted in Manhattan and Euclidean distance by changing the value of p, if $p = 1$, then this formula is considered as Manhattan distance, if $p = 2$, this formula is considered as Euclidean distance [6].

$$\text{Distance} = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \tag{1}$$

4.2 Manhattan Distance

Manhattan distance can be calculated by putting $p = 1$ in Minkowski distance. Manhattan distance is basically usual when a grid-like structure is given and a distance to be calculated between two points [6].

$$\text{Distance} = \sum_{i=1}^{n} |x_i - y_i| \tag{2}$$

4.3 *Euclidean Distance*

Euclidean distance formula can be obtained by putting $p = 2$ in Minkowski distance formula. Euclidean distance is a very popular distance metric to calculate displacement (displacement).

$$\text{Distance} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

5 KNN Model Using Machine Learning Approach

To build a model using KNN algorithm using machine learning approach, anaconda navigator and Jupyter notebook are used. Different Python libraries are also used to build the model whose description is given below [7].

- **Pandas**: To read the data set, Python panda's library is used.
- **Matplotlib**: This Python library is used for creating different kinds of visualization like bar charts, line charts, etc.
- **Seaborn**: This Python library is used for creating different kinds of visualization with statistical approach.
- **Scikit-Learn**: This Python library is used for creating different models. This library has a very efficient model based on classification, regression and clustering.

Implementation of any model in machine learning required seven basic steps:

1. Collection of data set
2. Data preprocessing
3. Data visualization
4. Model creation
5. Prediction
6. Model validation.

5.1 *Collection of Data Set*

Here, iris data set is used for KNN classification download for UCI machine learning repository. Download URL is https://archive.ics.uci.edu/ml/datasets/iris. Total 150 observations are given in the data set with four features (sepal length, sepal width, petal length, petal width) and one target variable that is iris species. Classification of iris flowers is done on the basis of its sepal and petal. By measuring its sepal length, sepal width, petal length and petal width, iris flower is classified into its three species: 1. *Iris Setosa*, 2. *Iris Versicolour*, 3. *Iris Virginica* (Figs. 2 and 3).

Fig. 2 Iris flower

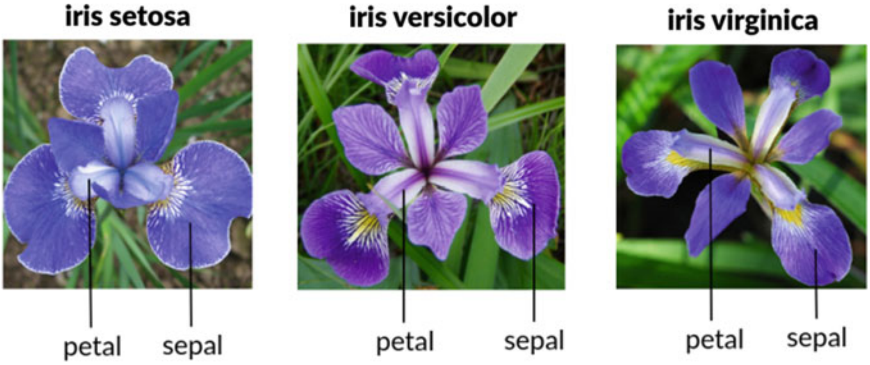

Fig. 3 Different species of iris flower

5.2 Data Preprocessing

Data preprocessing is the first step while adopting machine learning approaches. In data preprocessing, all relevant columns (features) can be considered. Each feature is further analysis for null value. If any null value is there, it should be removed [8]. Data set may not contain all columns in numeric form, there may be categorical variables, for example, male, female, but machine is not accepting these kinds of categorical variables. So, these categorical variables will be changed in numeric form [8–11].

Data normalization is another step in data preprocessing. Data is normalized around the mean value with standard deviation 1. This data normalization is also known as standard scalar.

5.3 Data Visualization

As shown in Fig. 4, we can see that each iris species is clearly separable based upon petal length, with just a bit of potential overlap in the *Iris-versicolor* and *Iris virginica*, while *Iris-setosa* is completely separable from the other two. It had been observed that statistically *Iris-setosa* is different from others too, while checking its parameters on mean median and mode value.

As shown in Fig. 5, we can see that each iris species is not clearly separable based upon sepal width, with huge overlap in the *Iris-versicolor*, *Iris virginica* and *Iris-setosa.* It had been observed that statistically iris species are very similar too, while checking its parameters on mean median and mode value.

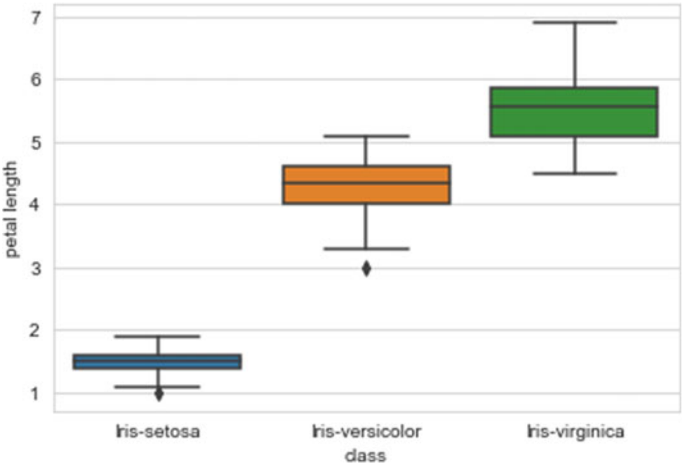

Fig. 4 Box plot between iris classes and petal length

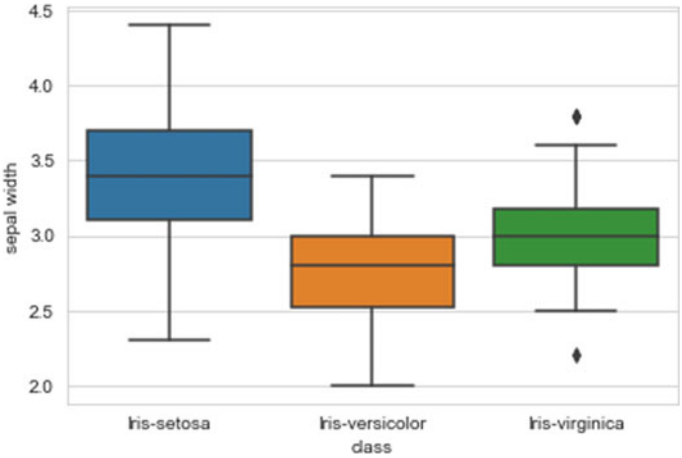

Fig. 5 Box plot between iris classes and sepal width

As shown in Fig. 6, we can see that each iris species is not clearly separable based upon sepal length, with just overlap in the *Iris-versicolor*, *Iris virginica* and *Iris-setosa.* It had been observed that statistically iris species are very similar too, while checking its parameters on mean median and mode value.

As shown in Fig. 7, we can see that each iris species is clearly separable based upon sepal length, with just overlap in the *Iris-versicolor*, *Iris virginica* and *Iris-setosa*, while *Iris-setosa* is completely separable from the other two.

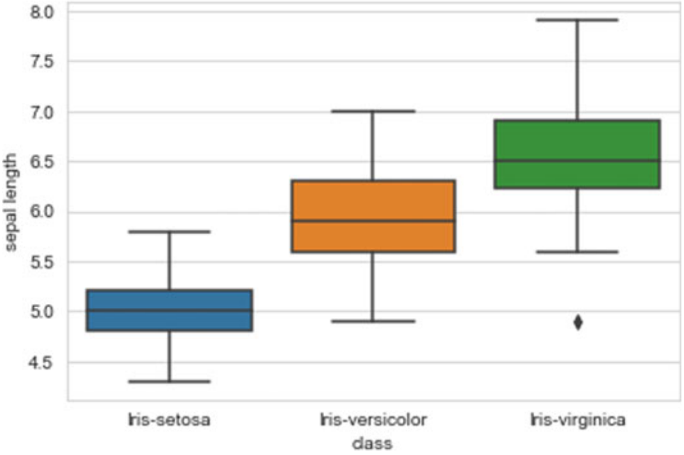

Fig. 6 Box plot between iris classes and sepal length

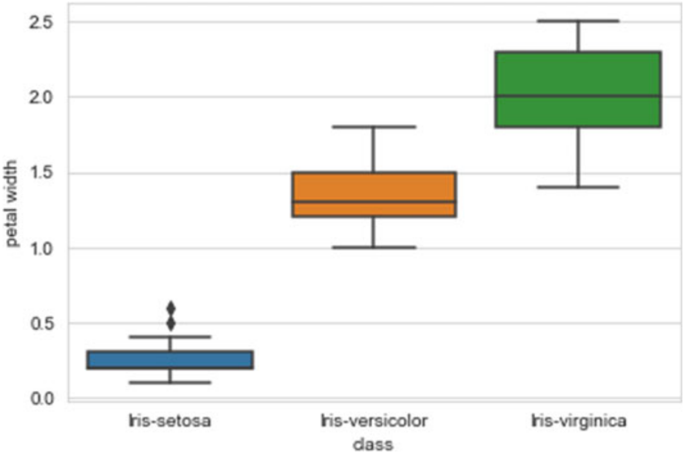

Fig. 7 Box plot between iris classes and petal width

Count plot in Fig. 8 is used to showing the total instance associated with each iris species *Iris-versicolor*, *Iris virginica* and *Iris-setosa*. In our data set, 50 data samples belong to each class.

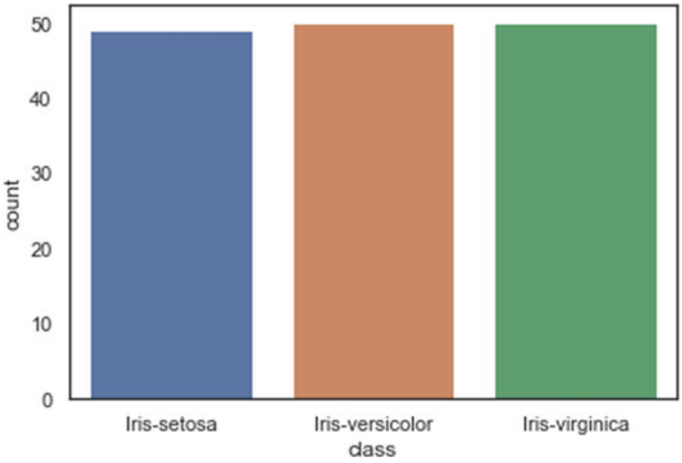

Fig. 8 Count plot between iris classes and total count of each species

Table 1 *K*-value and classification accuracy

k-value	Classification accuracy in %
1	95.0
2	95.0
3	95.0
4	98.3
5	96.6
6	98.3
7	98.3
8	98.3
9	98.3
10	98.3
11	98.3
12	98.3
13	95.0
14	96.0

5.4 Model Creation

For creating KNN model, Scikit-learn library is used. Data shuffling is an important step, so before fitting data to our algorithm, data is shuffled so that training on important features is distributed equally. With the help of training shuffled data, it is fit to the KNN model using fit() method. Once data gets fitted or the model gets trained, this will start prediction. The KNN model will predict the label for new data [12].

5.5 Prediction

Prediction is obtained from the KNN model by taking a different number of k values with distance metric as Euclidean. It is observed that initially classification accuracy is constant ($k = 1$ to 3) but after increasing the value of k it increases and maximizes ($k = 4$ to 12) after that classification accuracy is decreased [3–5] (Table 1; Fig. 9).

5.6 Model Validation

When performing model validation using training and testing accuracy, a fact is found out regarding number of neighbours that is accuracy is stable at $k = 10$ to 12

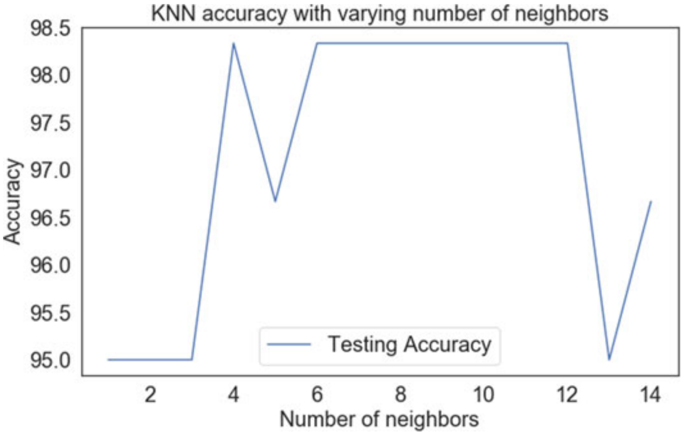

Fig. 9 Classification accuracy versus number of neighbour

in both cases training and testing. So for iris data set when performing K-nearest neighbour algorithm using machine learning approach, 10–12 number of neighbour give best/stable accuracy.

6 Conclusion

Iris data set is tested on K-nearest neighbour algorithm to find different number of neighbour value. Different number of K-nearest neighbour had been passed to model to check impact on classification accuracy. Besides this, ideally there must be an odd value of k in the range which gives maximized classification accuracy. In our result, set k value be 11 (get an odd number of neighbours). These values are odd and give maximized classification accuracy for our iris data set. These K value may be different from data set to data set, but in each data set case, iteration of K-value must be performed to find model accuracy.

References

1. Thirunavukkarasu K, Singh AS, Rai P, Gupta S (2018) Classification of IRIS dataset using classification based KNN algorithm in supervised learning. In: 4th international conference on computing communication and automation. https://doi.org/10.1109/CCAA.2018.8777643
2. Kulkarni SB, Kulkarni RB, Kulkarni UP, Hegadi RS (2014) GLCM-based multiclass iris recognition using FKNN and KNN. Int J Image Graph 14(3):1450010 (27 pages). https://doi.org/10.1142/S0219467814500107
3. Tan S (2005) Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Syst Appl 28:667–671

4. Hastie T, Tibshirani R (1996) Discriminant adaptive nearest neighbor classification. IEEE Trans Pattern Anal Mach Intell 18:607–616
5. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244
6. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
7. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
8. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
9. Sharma KP, Poonia RC, Sunda S (2019) Map matching algorithm: curve simplification for Frechet distance computing and precise navigation on road network using RTKLIB. Cluster Comput 22(6):13351–13359
10. KP Sharma, RC Poonia (2018) Review study of navigation systems for Indian regional navigation satellite system. (IRNSS) Soft Comput Theor Appl 735–742
11. Sharma KP, Poonia RC, Sunda S (2017) Map matching approach for current location tracking on the road network map matching approach for current location tracking on the road network. In: International conference on infocom technologies and unmanned systems, pp 573–578.
12. https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f