Study of distance metrics on k - nearest neighbor algorithm for star categorization

Swathi Nayak¹, Manisha Bhat¹, N V Subba Reddy¹, B Ashwath Rao¹
¹Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India-576104

Abstract. Classification of stars is essential to investigate the characteristics and behavior of stars. Performing classifications manually is error-prone and time-consuming. Machine learning provides a computerized solution to handle huge volumes of data with minimal human input. k-Nearest Neighbor (kNN) is one of the simplest supervised learning approaches in machine learning. This paper aims at studying and analyzing the performance of the kNN algorithm on the star dataset. In this paper, we have analyzed the accuracy of the kNN algorithm by considering various distance metrics and the range of k values. Minkowski, Euclidean, Manhattan, Chebyshev, Cosine, Jaccard, and Hamming distance were applied on kNN classifiers for different k values. It is observed that Cosine distance works better than the other distance metrics on star categorization.

1. Introduction
Stars are classified based on their spectral characteristics. To study what stars are made up of, to measure the hotness of stars, and to determine the evolution stage of a star, spectral classification of stars is essential. Due to the sheer quantity of data, it is not feasible for human experts to manually classify them. The separation of stars and galaxies into different categories needs to be automated. Machine learning is a technique for information investigation that computes scientific model structure. It is a part of computerized reasoning dependent on the possibility that frameworks can gain from information, recognize examples and settle on choices with insignificant human mediation. It is practically difficult to settle on significant decisions by utilizing manual examination of data sets that have gigantic information. Data Mining has made it conceivable to dissect information proficiently from various viewpoints. It is simpler to sort the information using different methodologies of data mining. Presently due to the number of methods available to mine the data it is feasible to process varieties of data, sum up the data, and discover the relationship among different data.

Data Mining approaches are broadly classified as supervised and unsupervised. One such supervised learning approach is the k-Nearest Neighbor (kNN) algorithm. For the calculation to work best in kNN on a specific dataset we need to pick the most suitable distance metric and k value appropriately. There are various distance measurements accessible. The main aim of this work is to compute the accuracy of the kNN algorithm using different distance metrics and ranges of k values for star categorization.

2. Literature survey
Chomboon K et al. [1] in their work have analyzed the performance of the kNN algorithm by comparing the different distance metrics. In this paper, 11 different distance metrics were studied on eight different datasets. Jaccard and Hamming distance techniques were seen to be affected by the
amount of data present in each category class. Cityblock, Euclidean, Mahalanobis, Minkowski, Chebyshev, and Euclidean were seen to be showing higher classification accuracy.

Yean C W et al. [2] in their work have aimed at studying the performance of eight different distance metrics using the kNN classification algorithm. kNN classifier is used to classify the emotional Electroencephalogram (EEG) signal of stroke and normal patients. The EEG signals were analyzed in different frequency bands (alpha, beta, and gamma). The value of k in the kNN algorithm was varied in the range of 1 to 15 and the accuracy of classification obtained against each of the k values was studied. City block distance metric performed the best in the beta band with the highest classification accuracy.

Ganesan K et al. [3] in their work have classified the Magnetic Resonance Imaging (MRI) images into normal and abnormal tissue images based on their textural features. First-order statistical features and segmentation-based fractal texture-based analysis classification of MRI images was carried out using a kNN - classifier. The performance of the kNN classifier was analyzed by applying various distance metric techniques like Euclidean, City block, Correlation, and Cosine. The classification resulted in higher accuracy for First-order statistical features when the Euclidean distance metric was used.

Moldagulova A et al. [4] in their work have proposed the kNN algorithm for the classification of textual documents into predefined categories. The value of k was varied and its corresponding classification accuracy was studied. Accuracy of text classification was found to be higher for values of k less than or equal to 50 and the accuracy was seen to drop sharply for values of k above 50.

Mulak P et al. [5] in their work have aimed at comparing Euclidean distance, Chebyshev distance, and Manhattan distance metrics using the kNN algorithm on Knowledge Discovery in Databases (KDD) data set. Data is preprocessed which includes normalization, data cleaning, and data integration. The performance of the classifier against each of the distance metrics is verified based on the accuracy of classification, specificity, sensitivity, false-positive rates, and false-negative rates. Out of the three distance metrics used, Manhattan was found to give higher performance followed by Chebyshev and then the Euclidean distance metric. Thirunavukkarasu K et al. [6] have classified the iris dataset using a kNN classifier.

Vaishannave M P et al. [7] have classified the groundnut leaf diseases into four major classes using the kNN classification algorithm. By suitably selecting the value of k, which resulted in higher accuracy, classification of leaf diseases was achieved. Jivani A G [8] in her work has categorized the text data using the novel k nearest algorithm. The value of the k chosen was based on the size of the class with the least number of training instances. From each of the classes, 'n' nearest neighbors of the test instance were found, where 'n' is a value not greater than the size of the smallest class. This ensured that an equal number of nearest neighbors were selected from each of the classes. Then the k nearest neighbors were chosen from the set of all the 'n' nearest neighbors belonging to each of the classes and the test data was classified as belonging to one of the classes. The results obtained showed that a higher value of k gave accurate output.

Saxena K et al. [9] in their work have aimed at the prediction of diabetes mellitus using the kNN algorithm. The accuracy and error rate of the prediction was tested for k values of 3 and 5. The results showed that the accuracy of the prediction and the error rate increased with the increase in the value of k. Rodrigues E O [10] in his paper has proposed a distance measure, that combines Chebyshev and Minkowski distance metric for the kNN classification algorithm. The kNN algorithm with the proposed distance measure was applied on 33 different datasets. The value of k was varied from 1 to 200 and the accuracy of classification was tested for the proposed distance metric and also other distance metrics like Manhattan, Euclidean, etc.

Dragomir E G [11] in the proposed work, has aimed at the prediction of the Air Quality Index using the kNN algorithm. Euclidean distance measure was used to calculate the distances between the test and the training data. The dataset considered, had instances of air composition, recorded for a shorter period, and hence it affected the accuracy of the prediction model. Baldini G et al. [12] in their work have aimed at detecting mobile malware using the kNN algorithm by emphasizing the effect of
various distance measures on the classification performance of the algorithm. The classification was
done by varying the value of k in the range of 1 to 9 in steps of two and nearly eight different distance
metrics, namely, Correlation, Jaccard, Euclidean, Minkowski, City block, Chebyshev, Hamming, and
Spearman were studied for their performance in the malware detection task. False Positive Rate was
the metric used for measuring the performances of all the distance metrics studied over a range of k
values.

Walters - Williams J et al. [13] in their work have compared the distance functions applied to the
nearest neighbor algorithm. Six distance metrics namely, Euclidean, Hamming, Kullback- Leibler,
Manhattan, Mahalanobis, and Minkowski were studied. Experimental results showed that the
Mahalanobis distance metric performed better compared to the other distance metrics studied in the
paper.

3. Dataset
The dataset [14] considered in this paper is the star dataset with 240 records. It has numerical features
namely, temperature, relative luminosity, relative radius, absolute magnitude, and categorical features
like star color, spectral class, and type. Based on the features mentioned, the star is categorized as
belonging to one of the six types namely red dwarf, brown dwarf, white dwarf, main sequence,
supergiants, and hypergiants. Neither duplicate values nor missing values are present in the dataset.

4. Methodology
kNN algorithm uses distance metric techniques to identify the similarities and dissimilarities between
the data points. The similarity between the data points increases with a decrease in the distance
between them. The kNN algorithm stores the training data and then uses these data instances to predict
the test data. From the training dataset, k nearest neighbors are determined, for every new test data
point. The algorithm works based on the concept of majority voting hence, the chosen test data point is
classified based on a majority of votes from its k neighbors in training data points.

In this paper, star categorization is done by applying various distance metrics to the kNN
algorithm. For each distance metric used, the K value is varied from 3 to 13 and that value of k, which
gives the maximum classification accuracy, is noted. Dataset is split into train and test sets with test
size $\beta = 0.3$. We have used KNeighborsClassifier for model training and predictions from an open-
source library called SCIKIT-LEARN [15] in python. Implementation steps are as mentioned in
algorithm 1.

Algorithm 1: Determining the maximum accuracy and corresponding k value for distance
metrics

| Input: | Separate data and class label in the dataset as X and Y respectively. Perform train_test_split on X and Y with test_size = $\beta$. Assign the values to x_train, y_train, x_test, and y_test |
|---|---|
| Output: | Maximum accuracy and corresponding optimal k value for distance metric |
| 1: | DISTANCE_METRICS = \{cosine, euclidean, chebyshev, minkowski, manhattan, hamming, jaccard\} |
| 2: | Set MAX_ACCURACY to an empty dictionary |
| 3: | for all D \in DISTANCE-METRICS do |
| 4: | Set KNN_ACCURACY to an empty dictionary |
| 5: | for k=3 to 13 step 2 do |
| 6: | KNN = KNeighborsClassifier(n_neighbors=k, metric=D) |
| 7: | Model = Apply KNN on x_train and y_train |
| 8: | Score = Apply Model on x_test and y_test |
| 9: | Insert k and Score onto KNN_ACCURACY as key-value pair |
| 10: | end for |
| 11: | Add D and maximum score of KNN_ACCURACY onto MAX_ACCURACY as key-value pair |
| 12: | end for |
The seven distance metrics applied to the classification of stars are discussed in the following section.

4.1. Minkowski Distance
Minkowski is a generalized distance metric, which can be used to calculate three different distance metrics namely, Manhattan distance, Euclidean distance, and Chebyshev distance by suitably setting the value of \( p \) in its distance formula. Let \( x = (x_1, x_2, \ldots, x_n) \) and \( y = (y_1, y_2, \ldots, y_n) \) be two vectors, where \( n \) is the number of variables then, the Minkowski distance formula is given by equation (1):

\[
D(x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}
\]

4.2. Euclidean Distance
The Euclidean distance measure is one of the commonly used distance metrics to calculate the distance between two vectors. It is calculated as the square root of the sum of squares of the differences between the cartesian coordinates of data points. Setting the value of \( p \) as 2 in the Minkowski distance formula, gives us the Euclidean distance between vectors \( x \) and \( y \) is given by equation (2):

\[
D(x, y) = \left( \sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}
\]

4.3. Manhattan Distance
Manhattan distance measure calculates a distance between two data points as a sum of the absolute differences of their cartesian coordinates. When the data points have higher dimensions, then the Manhattan distance metric is preferred to the Euclidean distance metric. Setting the value of \( p \) as 1 in the Minkowski distance formula, gives us the Manhattan distance between vectors \( x \) and \( y \) is given by equation (3):

\[
D(x, y) = \sum_{i=1}^{n} |x_i - y_i|
\]

4.4. Chebyshev Distance
Chebyshev distance, also known as chessboard distance, measures the distance between the data points as the maximum difference between the coordinates of the data points. Chebyshev distance is obtained by substituting the value of \( p \) as infinity in the Minkowski distance formula. Chebyshev distance between two vectors \( x \) and \( y \) are given by equation (4):

\[
D(x, y) = \max_i |x_i - y_i|
\]

4.5. Cosine Distance
The cosine distance metric measures the degree of angle between vectors. It is generally used when the orientation of the vector is of concern and not its magnitude. The values of the cosine angles are looked for to find similarities between the vectors. If the cosine angle value is 1, then the data points are oriented in the same direction and hence are similar. If the cosine value is -1 then, the data points are oriented in opposite directions and dissimilar. If the cosine value is 0, then there might be some similarity between the data points. Cosine distance is calculated as one minus the Cosine similarity coefficient. The higher the value of the Cosine similarity coefficient between two vectors higher is the similarity between them. Cosine distance formula for two vectors \( A \) and \( B \) is given by equation (5):

\[
D(A, B) = 1 - \frac{A \cdot B}{\|A\|\|B\|}
\]
4.6. Jaccard Distance
Jaccard similarity coefficient is used for measuring the similarity between finite sample sets. The higher the value of the Jaccard similarity coefficient for two sets of data, the higher is the similarity between them. Jaccard distance is calculated as one minus the Jaccard similarity coefficient. Jaccard distance for sets A and B is calculated as given by equation (6):

$$D(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$ (6)

4.7. Hamming Distance
Hamming distance is used to calculate the distance between the data points when all of its features have binary values i.e. it compares the binary strings representing features of each of the data points. As a part of the data preparation step, the categorical variables in a feature column should be suitably encoded into binary vectors for hamming distance measurement. Hamming distance between two data points x and y are given by equation (7):

$$D_H = \sum|X_i - Y_i|$$ (7)

If x = y then D = 0
If x ≠ y then D = 1

5. Results
In this paper, we have compared the accuracy of the seven distance metrics namely Minkowski, Euclidean, Manhattan, Chebyshev, Cosine, Jaccard, and Hamming distance for the classification of stars into six broad categories using the kNN algorithm. The value of k was varied in the range of 3 to 100, to find that value of k, which served better classification accuracy. For all the distance metrics considered in this paper, the maximum value of classification accuracy found was for k values in the range 3 to 9. The plot for the accuracy values corresponding to the k values in the range 3 to 13 for each of the distance metrics, is shown in figure 1-7.

![Figure 1. A plot of k value v/s Accuracy for Minkowski distance metric](image-url)
Figure 2. A plot of k value v/s Accuracy for Euclidean distance metric

Figure 3. A plot of k value v/s Accuracy for Manhattan distance metric

Figure 4. A plot of k value v/s Accuracy for Chebyshev distance metric

Figure 5. A plot of k value v/s Accuracy for Cosine distance metric

Figure 6. A plot of k value v/s Accuracy for Jaccard distance metric

Figure 7. A plot of k value v/s Accuracy for Hamming distance metric
Since values of $k$ beyond 10 didn’t change the maximum value of classification accuracy, accuracy values mentioned in table 1 are limited for values of $k$ ranging between 3 to 13. The average accuracy of the seven distance metrics ranged between 25% to 85%, when checked over a range of $k$ values. Cosine distance metric has higher accuracy at $k$ value of 9. Minkowski, Euclidean, Manhattan, and Chebyshev have higher accuracy at a $k$ value of 5. Hamming has optimum accuracy at a $k$ value of 3. Jaccard distance showed constant accuracy for all values of $k$. Hence all the seven distance metrics considered showed maximum classification accuracy for values of $k$ below 10.

Cosine Distance metric performed the best amongst all the distance metrics and gave maximum accuracy of 84.72% and Jaccard performed the least with an accuracy of 25%. Maximum accuracy for each of the distance metrics is shown in table 2.

**Table 1.** Table showing the accuracy of the distance metrics corresponding to the $k$ values in the range 3 to 13.

| $k$ | Manhattan | Euclidean | Chebyshev | Minkowski | Cosine | Jaccard | Hamming |
|-----|-----------|-----------|-----------|-----------|--------|---------|---------|
| 3   | 59.72%    | 58.33%    | 59.72%    | 58.33%    | 82.00% | 25.00%  | 61.11%  |
| 5   | 66.66%    | 62.50%    | 62.50%    | 62.50%    | 83.33% | 25.00%  | 57.00%  |
| 7   | 55.55%    | 54.16%    | 55.55%    | 54.16%    | 83.33% | 25.00%  | 57.00%  |
| 9   | 52.77%    | 51.38%    | 54.16%    | 51.38%    | 84.72% | 25.00%  | 51.38%  |
| 11  | 52.77%    | 51.38%    | 54.16%    | 51.38%    | 80.55% | 25.00%  | 51.38%  |
| 13  | 50.00%    | 50.00%    | 52.77%    | 50.00%    | 79.16% | 25.00%  | 51.38%  |

**Table 2.** Table showing maximum accuracy of distance metrics for star categorization.

| Distance Metrics | $k$ | Maximum accuracy |
|------------------|-----|------------------|
| Manhattan        | 5   | 66.66%           |
| Euclidean        | 5   | 62.50%           |
| Chebyshev        | 5   | 62.50%           |
| Minkowski        | 5   | 62.50%           |
| Cosine           | 9   | 84.72%           |
| Jaccard          | 3   | 25.00%           |
| Hamming          | 3   | 61.11%           |

6. Conclusion

In this paper, we have aimed at analyzing the performance of different distance metrics using the kNN classification algorithm on the star dataset. The accuracy of seven distance metrics was analyzed by varying the values of $k$ over a range. Cosine distance performed the best for the star dataset and Jaccard performed the least amongst all the distance metrics.

It was observed that even if the range of $k$ for the classification of stars was increased, the value of $k$ which yielded higher classification accuracy didn’t show significant change. It was seen that higher values of $k$ showed local maxima values in terms of accuracy but, the global maxima value for accuracy remained unchanged.

As a future work, instead of the kNN algorithm for classification, weighted kNN can be considered for better accuracy. Efficiency assessment of the model can be done by cross-validation using different datasets. Many other distance metrics can also be studied, to find out if there is any distance metric that could give better accuracy than Cosine, and also if there is any distance metric that performed poorer than Jaccard when applied on the star dataset. These additions would help in better understanding the effects of distance metrics on the classification of the stars.
References:

[1] Chompoon K, Chujai P, Teerarassamee P, Kerdprasop K and Kerdprasop N 2015, March. An empirical study of distance metrics for the k-nearest neighbor algorithm. In Proc. of the 3rd Int. Conf. on Industrial Application Engineering, pp 280-285

[2] Yean C W, Khairunizam W, Omar M I, Murugappan M, Zheng B S, Bakar S A, Razlan Z M and Ibrahim Z 2018, August. Analysis of the distance metrics of kNN classifier for EEG signal in stroke patients. In 2018 Int. Conf on Computational Approach in Smart Systems Design and Applications, pp 1-4

[3] Ganesan K and Rajaguru H 2019. Performance analysis of kNN classifier with various distance metrics method for MRI images. In Soft Computing and Signal Processing, pp 673-682

[4] Moldagulova A and Suliaman R B 2017, Using kNN algorithm for classification of textual documents. In 2017 8th Int. Conf. on Information Technology, pp 665-671

[5] Mulak P, Talhar N 2015, Analysis of distance measures using k-nearest neighbor algorithm on KDD dataset. Int. J. Sci. Resi., pp 2101-2104

[6] Thirunavukkarasu K, Singh A S, Rai P, Gupta S 2018, December. Classification of iris dataset using classification based kNN algorithm in supervised learning. In 2018 4th Int. Conf on Computing Communication and Automation, pp 1-4

[7] Vaishannave M P, Devi K S, Srinivasan P, Jothi G A P 2019, March. Detection and classification of groundnut leaf diseases using kNN classifier. In 2019 IEEE Int. Conf. on System, Computation, Automation and Networking, pp 1-5

[8] Jivani A G 2013, January. The novel k nearest neighbor algorithm. In 2013 Int. Conf. on Computer Communication and Informatics, pp 1-4

[9] Saxena K, Khan Z, Singh S 2014. Diagnosis of diabetes mellitus using k nearest neighbor algorithm. Int. J. of Computer Science Trends and Technology, vol 2, pp 36-43

[10] Rodrigues E O 2018. Combining minkowski and chebyshev: new distance proposal and survey of distance metrics using k-nearest neighbors classifier. Pattern Recognition Letters, vol 110, pp 66-71

[11] Dragomir E G 2010. Air quality index prediction using k-nearest neighbor technique. Bulletin of PG University of Ploiesti, Series Mathematics, Informatics, Physics, vol LXII, no. 1, pp 103-108

[12] Baldini G, Geneiatakis D 2019, April. A performance evaluation on distance measures in KNN for mobile malware detection. In 2019 6th Int. Conf. on control, decision and information technologies, pp 193-198

[13] Walters–Williams J, Li Y 2010. Comparative study of distance functions for nearest neighbors. In Advanced techniques in computing sciences and software engineering, pp 79-84

[14] https://www.kaggle.com/brsdincer/star-type-classification

[15] Kramer O, 2016. Scikit-learn. In Machine learning for evolution strategies, pp 45-53