Knn – K Nearest Algorithm

Student ID

Name

Date

1. Introduction

K-nearest neighbour (k-NN) algorithm is among the simplest and most intuitive algorithms in machine learning. It is conceptually simple, but it carries a number of significant conceptualizations regarding distance, similarity, and geometric organization of data. Being a non-parametric and instance-based learning method, k-NN is not based on a pre-defined functional mapping between features and labels. Rather, it gives predictions of the association of a new data point with the previous examples in the training set. The aim of this tutorial is to present a detailed description of the k-NN algorithm, as well as to illustrate the effect of its most important hyperparameter, k on the performance of classification. As a result of this demonstration, the readers will be aware of such concepts like the bias- variance trade-off, overfitting, underfitting, and the significance of preprocessing data. The Wine data sample, which is provided by Scikit-Learn, is utilized as a realistic case due to its medium size, highly numeric characteristics, and structured classification (Jodas, 2023). The tutorial by investigating the behaviour of k-NN on this dataset offers a clear and well-organised route towards understanding the theoretical and practical factors of neighbourhood-based classification.

2. Knowledge of k-Nearest Neighbour Algorithm.

The k-NN algorithm has a very simple yet effective premise which states that the points that are in close proximity with one another in the feature space are likely to have similar labels. In order to classify the new, unobserved sample, the algorithm uses the nearest k samples in the training set to classify the unknown case with the label that the most number of samples have. The Euclidean distance metric is used most often in the determination of distance, and is the straight-line distance between two points in multi-dimensional space. Since k-NN is a completely distance-based technique, feature scaling is an important part of the process. When one feature takes on an enormously larger numerical range than the other, then it will have disproportionate effect on the calculation of distance, which may result in erroneous neighbour relationships and poor classification.

The main feature of k-NN is to select the parameter k, which refers to the number of neighbours that are taken into account when making the prediction. In the case when k is one, the algorithm is very flexible, which is virtually memorising the training data. This makes it very sensitive to noise and outliers and enables it to model very complex patterns. One mislabeled or noisy point can have a significant effect on the outcome leading to high variance and inability to classify new data. On the other hand, with a very large k, the classifier becomes too smooth, smoothing

over a large number of points although they may be of different classes. This causes a lot of bias and may underfit (Martín-Martín, 2023). The perfect value of k is neither a small nor a big number, and its choice will play a vital role in having a balanced performance. The combination of flexibility and stability is the cornerstone of the bias-variance trade-off, which is a core concept of machine learning.

3. The Wine Dataset

The data in this tutorial is the Wine dataset which was initially downloaded in the UCI Machine Learning Repository and is provided in Scikit-Learns built-in datasets. It has 178 samples of wine based on three cultivars, which are each characterized by 13 continuous chemical measures. These characteristics are alcohol content, malic acid, magnesium, phenols, flavanoids, colour intensity and many others that characterize chemical profile of the wine. The dataset is also balanced in terms of its three classes, whereby it has sufficient samples of the three classes to enable effective model evaluation. Since the dataset only has numerical characteristics of different magnitudes, it gives a viable example of why scaling is necessary in the distance-based approach like k-NN. Also, the dataset does not have any missing values or categorical attributes that makes the preprocessing easier and makes one able to focus directly on the behaviour of the algorithm itself (Bullejos, 2022).

Wine is frequently included in the introductory machine learning tutorials due to its size and sufficient size allowing an efficient analysis in addition to providing a realistic perspective on chemical, biological and sensory classification tasks. Advanced understanding of multi-dimensional patterns interpretation by models is also supported by the numerical richness of the dataset, which makes it the best case to exemplify classification methods and hyperparameter optimization.

4. Methodology

This tutorial is structured in a systematic approach to the analysis of the behaviour of k-NN with Wine data. The preliminary stage is to load the dataset with the load_wine() function of Scikit-Learn that gives the feature matrix and labels. The data is further divided into training and test data by stratified 70 /30 split to maintain the proportional distribution of classes (Islam, 2022). This will make sure that the model which is being trained is sufficiently exposed to all classes and that the test set is a reasonable representation of the composition of the data.

The second important step is feature scaling. Since the data on wine involves chemical measurements of different numbers, it is necessary to use a standardisation process with the

Scikit-Learn StandardScaler. The standardisation makes every feature mean, with the standard deviation of zero and one. This normalises the size of all the dimensions and the Euclidean distance computation is not dominated by a single feature. In the absence of such transformation, k-NN would prefer to use features that are inherently larger leading to bias and inaccurate predictions (Ghosh, 2022).

After preprocessing, the k-NN models are trained using a parameter of variation k between 1 and 30. The model is fitted on the scaled training data, and its predictive accuracy of the model is reported on the training and the test data, given each value of the k in the model. These are measured and used subsequently in order to generate an accuracy curve, which demonstrates the variation of model performance with an increase in k. This methodical analysis allows seeing the behaviour of the model in detail and emphasises the practical consequences of choice of various values of k. The last element of the methodology is the determination of the most efficient k value according to the test accuracy and analysis of the accuracy curve (Anggoro, 2021).

5. Results

The analysis of k-NN models by training and evaluating them with various values of k shows a definite trend and matches the theoretical one. In the case of k=1, the model attains a perfect fit to the training data. This result is a consequence of the fact that every training point is the closest to itself, which makes it possible to memorise all the data using the model. This does not however translate into a strong performance on the test set. Test accuracy in small values of k is usually lower due to the fact that the model is overfitting the noise and idiosyncrasies in the training data.

The performance in training accuracy decreases slowly as k increases, indicating the more conservative and smooth boundaries of decision making by larger neighbourhoods. Remarkably, the accuracy of the tests also rises with the value of k going through small, moderate values implying that the model is generalising well. The test accuracy in this region of equalisation which is usually between a k of eight and twenty is constant and highest. The accuracy of the test is 100% in the case of the Wine dataset at k of approximately twenty-one. Even though it cannot be assured that the dataset is perfectly testable in the real-life situation, the structure of the Wine dataset and the separability of its classes does enable a very good performance of the model under the condition that the model is tuned accordingly (Yulianti, 2022).

Increasing values of k, there are overall strong test accuracy fluctuations. These swings suggest that, whereas bigger neighbourhoods are still producing predictable results, gross smoothing can blur smaller class-specific differences. However, the findings confirm that moderate k values are the ones where bias and variance are the most balanced, which underscores the fact that this parameter needs to be tuned.

6. Discussion

The tendency in the findings gives a clear demonstration of the bias-variance trade-off, which is a major concept of machine learning. In cases where k is not large, the model has low bias since it can fit the training data. The variance is however high and this implies that the predictions made by the model can vary drastically with a slight change in the data. The result of this instability is poor generalisation as is observed by the poor test accuracy when k is small. On the other hand, in cases where k is too large the model will be highly biased since it becomes too rigid and makes use of too many neighbours to make every prediction. This reduces the sensitivity of the model to local structure and finer details, resulting in underfitting. This space between these extremes is the best place to be, and it is here that the model is sufficiently flexible to represent meaningful patterns, yet sufficiently stable to generalise appropriately (Anraeni, 2022).

The second significant point, which the results reveal, is the need to scale the features. Algorithms that are distance based like k-NN are purely based on geometrical connections between points. Such connections are distorted when story components possess different levels of numeric numbers. Standardisation makes each feature make an equal contribution allowing the model to find meaningful neighbourhood structure in the feature space. Even the most appropriate value of k would not give the strong results without the scaling process, which shows how the preprocessing decisions can have a rather serious impact on the performance of the classifiers.

The format of the Wine data also sheds some light on the strength and weaknesses of k-NN. Its medium size and lack of outliers render it perfect in neighbourhood classification. The numerical consistency of the chemical measurements is amenable to the Euclidean distance metric particularly when scaled. Nonetheless, the algorithm has the drawback that it is dependent on the storage of the entire training set, which points to a larger limitation of the algorithm being computationally infeasible when dealing with large datasets since it requires the calculation of distances to all training points on-the-fly at the time of prediction (Kim,

2021). This weakness does not apply to the present case but is relevant when using k-NN in real-life situations.

Even in relatively simple classification problems, the issue of ethical considerations is becoming more and more topical in machine learning. Even though the Wine data is not sensitive and not related to any personal traits that are considered sensitive, the ethical AI tenets remain applicable. One such principle is transparency and k-NN has an obvious benefit in this regard since its predictions can be directly related to individual training examples. Another important factor is fairness (Güvenç, 2021). In real world data, demographic or sensitive variables can be important in classification and k-NN can be biased by those examples that are stored and directly used in the training data. There is also accountability, which means that organisations using classifiers have to be able to defend and account the predictions. Since k-NN involves stored data to make predictions, it can be used to inspect carefully the origins of decisions. Lastly, the privacy should be put into consideration since k-NN would need that the entire training examples should be stored and they might have personal information related to the individual. Anonymisation of data and safe storage is thus necessary in the event of using this algorithm in sensitive applications (Ab Wahab, 2021).

7. Conclusion

This tutorial has discussed the k-Nearest Neighbour algorithm both conceptually and empirically, in terms of the impact of the selection of k on Wine dataset model performance. It has empirically illustrated the empirical spike of the trade-off between bias and variance as well as the fact that neither extremely small nor extremely large values of k are optimal. K moderate values, instead, find the balance to enable the classifier to generalise well. The tutorial has also emphasized the significance of the feature scaling in distance based learning and how making preprocessing choices might have a huge impact on predictive performance. The Wine data acted as a good test ground to illustrate these ideas because it is rich in a numerical sense, balanced and small enough.

Despite its simplicity, k-NN has useful behaviour that provides valuable understanding of classification methods, hyperparameter optimization, and data geometry. It is especially useful in the teaching of machine learning principles, and cases where transparency is necessary. Simultaneously, its weaknesses, including computational ineffectiveness on massive datasets and sensitivity to irrelevant features, teach us that we should always be conscious of the context and constraints of the problem under consideration to select the model. Through the knowledge

about the behaviour and mechanics of k-NN, students and practitioners can learn to appreciate the basic ideas of machine learning on which more advanced modelling methods are based.

References:

Jodas, D.S., Passos, L.A., Adeel, A. and Papa, J.P., 2023. PL-kNN: A Python-based implementation of a parameterless k-Nearest Neighbors classifier. Software impacts, 15, p.100459.

Martín-Martín, M., Bullejos, M., Cabezas, D. and Alcalá, F.J., 2023. Using python libraries and k-Nearest neighbors algorithms to delineate syn-sedimentary faults in sedimentary porous media. Marine and Petroleum Geology, 153, p.106283.

Bullejos, M., Cabezas, D., Martín-Martín, M. and Alcalá, F.J., 2022. A K-nearest neighbors algorithm in Python for visualizing the 3D stratigraphic architecture of the Llobregat River Delta in NE Spain. Journal of Marine Science and Engineering, 10(7), p.986.

Anggoro, D.A. and Aziz, N.C., 2021. Implementation of K-nearest neighbors algorithm for predicting heart disease using python flask. Iraqi Journal of Science, pp.3196-3219.

Islam, A., Belhaouari, S.B., Rehman, A.U. and Bensmail, H., 2022. K Nearest Neighbor OveRsampling approach: An open source python package for data augmentation. Software Impacts, 12, p.100272.

Itoo, F., Meenakshi and Singh, S., 2021. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. International Journal of Information Technology, 13(4), pp.1503-1511.

Ghosh, S., Singh, A., Jhanjhi, N.Z., Masud, M. and Aljahdali, S., 2022. SVM and KNN Based CNN Architectures for Plant Classification. Computers, Materials & Continua, 71(3).

Yulianti, D.R., Triastomoro, I.I. and Sa'idah, S., 2022. Identifikasi Pengenalan Wajah Untuk Sistem Presensi Menggunakan Metode Knn (K-Nearest Neighbor). Jurnal Tekinkom (Teknik Informasi dan Komputer), 5(1), pp.1-10.

Anraeni, S., Melani, E.R. and Herman, H., 2022. Ripeness identification of chayote fruits using HSI and LBP feature extraction with KNN classification. Ilk. J. Ilm, 14(2), pp.150-159.

Kim, J.H., Choi, J.H., Park, Y.H., Leung, C.K.S. and Nasridinov, A., 2021. Knn-sc: Novel spectral clustering algorithm using k-nearest neighbors. IEEE Access, 9, pp.152616-152627.

Ab Wahab, M.N., Nazir, A., Ren, A.T.Z., Noor, M.H.M., Akbar, M.F. and Mohamed, A.S.A., 2021. Efficientnet-lite and hybrid CNN-KNN implementation for facial expression recognition on raspberry pi. IEEE Access, 9, pp.134065-134080.

Güvenç, E., Çetin, G. and Koçak, H., 2021. Comparison of KNN and DNN classifiers performance in predicting mobile phone price ranges. Advances in Artificial Intelligence Research, 1(1), pp.19-28.