Impact of K on KNN Model Performance

1. Introduction

Applied for both classification and regression, K-Nearest Neighbours (KNN) is a supervised machine learning method. It is predicated on the similarity concept, wherein the nearest neighbours in the feature space define the categorisation or prediction of a new data point. KNN is flexible and simple to use as it is non-parametric—that is, it does not assume anything about the underlying data distribution. The method detects patterns based on proximity utilising distance measurements such as Euclidean or Manhattan distances by considering a predetermined number of neighbours (K). The steps involved in the classification are as follows:

- Represent data as points in an N-dimensional feature space.
- Choose the number of neighbours (K) that strikes a compromise between large values of K is smoothing projections and small values of K is sensitivity to noise.
- Calculate distances between the new data point and every dataset point using either Euclidean or Manhattan distance.
- Sort all the datapoints by distance then choose the closest K points.
- Assign the class with the majority vote among the K neighbours the new point.

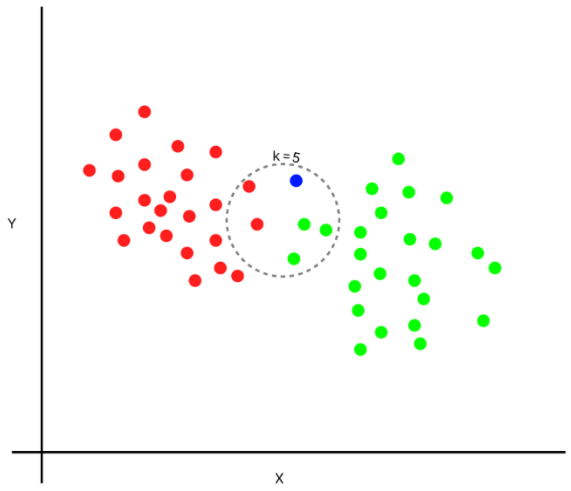


Figure 1 - KNN with 5 neighbours (Rishith, 2023)

2. The Role of K in Classification

An important decision that greatly affects the accuracy and performance of the model is selecting the value of K in the K-Nearest Neighbours (KNN) method. The number of neighbours used to classify a new data point is determined by the parameter value of K. Choosing the appropriate value requires balancing the ability to generalise well to unprocessed data with sensitivity to local trends.

Small K value: This classification method becomes rather sensitive to local patterns in the data when K is small, say K=1. Overfitting, in which the model picks noise and outliers in the training data, therefore producing poor generalisation to new data, might follow from this. A new data point that falls close to an outlier, for example, can be misclassified as the algorithm mostly depends on one closest neighbour. Although small K values could show good performance on the training set, they are more prone to large variance, hence their performance will vary greatly with little changes in the dataset.

Large K value: Greater K values evaluate a wider range of neighbours, therefore reducing the impact of individual data points. This helps to level out noise and outliers, therefore strengthening the algorithm and reducing its sensitivity to random fluctuations. On the other hand, too high values of K might cause underfitting—that is, the model oversimplifies the categorisation by neglecting significant local patterns. Under these circumstances, the model could fail to reflect the subtleties of the data distribution and the decision limits get less flexibility.

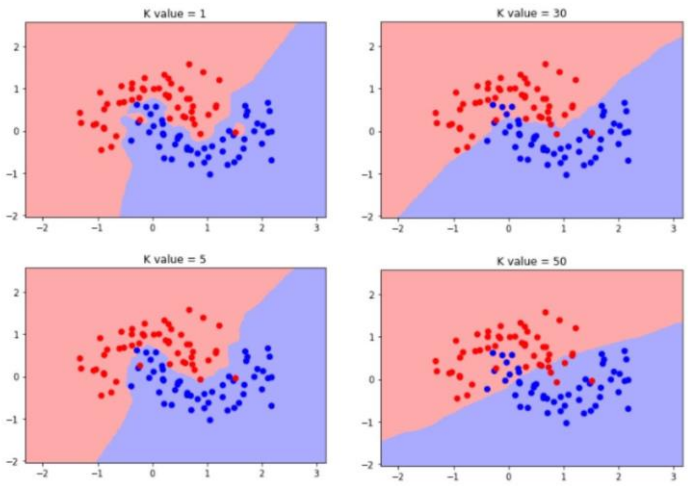


Figure 2 - Impact of decision boundary with the change in values of K (Kiran, 2022)

3. Choosing the Optimal K Value

Determining the ideal value of K usually requires validation and trial-run. These are many often used techniques for finding K:

- Cross-validation: One of the most used methods to choose K is by use of cross-validation. Multiple times the dataset is divided into training and validation sets; the model is tested on the validation set for varying K values. The best value is chosen as the K that produces the maximum validation accuracy.
- Odd Values of K: K is often used as an odd integer to prevent ties in voting, especially for binary classification jobs. This guarantees that among the neighbours there always is a clear majority class.
- Domain Knowledge: Sometimes the choice of K might be guided by information about the domain area.

4. Evaluating KNN Accuracy with Varying K

In this section, I will demonstrate how the K-Nearest Neighbours (KNN) algorithm performs varying values of K. Examining accuracy across a variety of K values will help to demonstrate the trade-offs between variation in KNN classification and bias. Using real-world code examples, I will demonstrate how although bigger values produce smoother, more broad predictions, lower values of K may lead to increased susceptibility to noise. This practical study will clarify the effect of K and advise in choosing the best value for KNN classification tasks.

Model Training:

```
k_range = [1, 3, 5, 10, 15, 20, 25, 50, 100, 250, 500]
train_scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(x_train, y_train)
    y_pred = knn.predict(x_train)
    accuracy = accuracy_score(y_train, y_pred)
    train_scores.append(accuracy)
    print(f"\nAccuracy of KNN classifier with K={k}: {accuracy * 100:.2f}%")
```

The KNN classifier's training accuracy shows an obvious pattern with variations in K. The model first obtains perfect accuracy of 100% with K=1. This is predicted as, with K=1, every data point is categorised depending on its closest neighbour, so it is rather likely to fit its own label. Such

great accuracy, however, points to overfitting—that is, a model that memorises the training data without effectively extending itself to fresh data.

The accuracy suffers to 90.01% when K rises to 3. By now the model is beginning to take more neighbours into account, which lessens the impact of individual data points and hence helps to avoid overfitting. A lower K value does, however, run the danger of the model being too sensitive to data noise, hence somewhat reducing accuracy. As K rises, the accuracy keeps becoming less; values of 5, 10, 15, and 20 indicate accuracies of 87.89%, 86.02%, 85.67%, and 85.28%, respectively. The model becomes broader by averaging across more neighbours as K rises, therefore producing a smoother decision border. This leads the model to lose some sensitivity to the particular patterns in the training data even as it lowers overfitting. The accuracy keeps steadily declining by the time K approaches 25 and above values (50, 100, 250, 500). This is so because the model basically loses sensitivity to the underlying structure of the training data with extremely large values of K, and the decision boundaries become too smooth. The classifier therefore loses sensitivity to the smaller differences in the training set, which somewhat reduces training accuracy.

Because the model forfeits sensitivity to individual data points in favour of smoother, more generalised decision boundaries, KNN's training accuracy generally declines as K grows. Choosing K means compromising between underfitting (with higher K) and overfitting (with smaller K). The ideal value for K achieves excellent performance on test and training data in balance.

Model Validation:

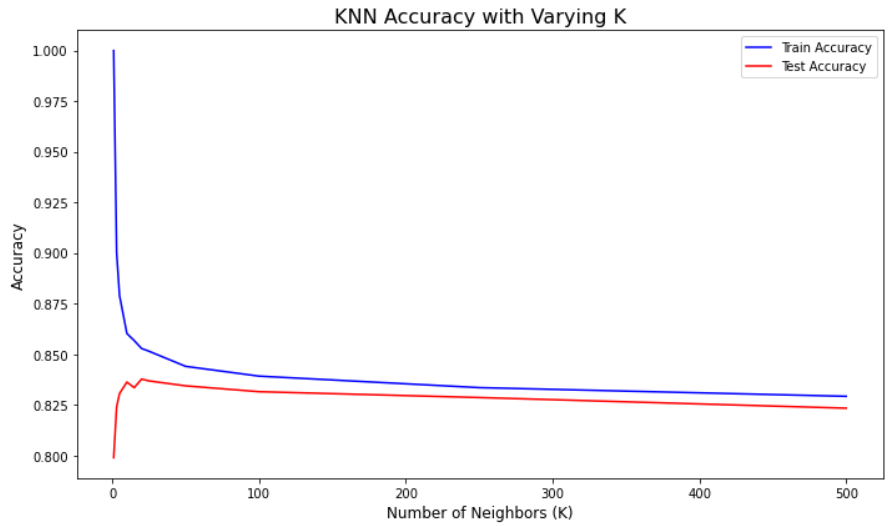
```
k_range = [1, 3, 5, 10, 15, 20, 25, 50, 100, 250, 500]
test_scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(x_train, y_train)
    y_pred = knn.predict(x_test)
    accuracy = accuracy_score(y_test, y_pred)
    test_scores.append(accuracy)
    print(f"\nAccuracy of KNN classifier with K={k}: {accuracy * 100:.2f}%")
```

Different values of K affect the test accuracy of the KNN classifier, therefore reflecting the sensitivity of the model to the number of neighbours it considers during classification. The model has more inclination to overfit the training data when K is small, say 1, which results in more influence of noise on forecasts. Here the accuracy is 79.89%, meaning that a lower K

produces a more volatile model with higher volatility. The accuracy rises to 82.40% as K moves to 3, implying that by smoothing out noise, a somewhat bigger neighbourhood helps the model generalise better.

With the maximum accuracy (83.62%), test accuracy shows constant gains in further increments in K to values as 5, 10, and 15. This implies that the model finds a reasonable equilibrium between underfitting and overfitting within this range. Although they may start to smooth out important patterns in the data, larger K values lower sensitivity to outliers and noise. Accuracy somewhat declines to 83.35% at K=15 and rises once more to 83.77% at K=20. This represents the model's attempt—with declining returns—to absorb more neighbours and lower variance. The accuracy gradually drops as K rises; for example, with K=50, 100, 250, and 500 the lowest test accuracy of 82.33% results. This decline is probably the result of the model become too generic. The model takes a wide range of neighbours at extremely high values of K, which might include distant or useless sites, thereby producing a less accurate categorisation. Generally speaking, the ideal number for K seems to be between 10 and 20 as the model maintains a reasonable balance between variance and bias.

Comparison:



Reflecting a balancing between overfitting and underfitting, the values of K in KNN have a major influence on both training and validation performance. The model indicates overfitting by reaching perfect training accuracy of 100% with K=1. The model therefore memorises the training data, which produces great performance on the training set but poor generalisation to new data. The model generalises better when K rises to 3; the accuracy falls to 90.01%.

Higher K values cause the model to become more generic, which reduces sensitivity to individual data points and so causes a consistent drop in training accuracy.

Regarding validation, the test accuracy is lower at 79.89% when K is small—that is, K=1—that indicates the model overfits the training data and is excessively sensitive to noise. Test accuracy rises to 82.40% when K rises to 3; it keeps rising, peaked at 83.62% for K=10. This indicates that the model gains from thinking through additional neighbours, noise and outlier smoothing, and balancing bias with variance. But as K rises over 20, the test accuracy begins to drop; at K=500, it reaches a low of 82.33% reflecting the model being too general and unable to detect meaningful trends.

K influences both training and validation performances; a lower K results in overfitting while a higher K causes underfitting. Between 10 and 20 is the ideal range for K as the model finds a decent trade-off between variance and bias.

5. Conclusion

The results of this tutorial highlight the important correlation between model performance in KNN classification and K value. The findings show that lower values of K (e.g., K=1) cause overfitting, in which case the model fits too closely to the training data, hence generating excellent training accuracy but poor generalising to new, unseen data. The model generalises better as K rises; test accuracy rises up to a point (K=10 to K=20) thereby balancing variance and bias. Further increases in K, however, lead to underfitting, in which case the model loses sensitivity to significant patterns and becomes too generalised, therefore reducing test accuracy. Consequently, the research shows that the choice of K directly affects the performance of the model; so, obtaining excellent prediction accuracy requires the identification of an ideal K and helps to prevent both overfitting and underfitting.

References

Beckmann, M., Ebecken, N.F. and Pires de Lima, B.S. (2015) 'A KNN undersampling approach for data balancing', Journal of Intelligent Learning Systems and Applications, 07(04), pp. 104–116. doi:10.4236/jilsa.2015.74010.

Bicego, M. et al. (2022) 'Advanced KNN approaches for explainable seismic-volcanic signal classification', Mathematical Geosciences, 55(1), pp. 59–80. doi:10.1007/s11004-022-10026-w.

Hou, J. et al. (2016) 'Feature combination and the KNN framework in Object Classification', IEEE Transactions on Neural Networks and Learning Systems, 27(6), pp. 1368–1378. doi:10.1109/tnnls.2015.2461552.

Nutanong, S. et al. (2008) 'The v*-diagram', Proceedings of the VLDB Endowment, 1(1), pp. 1095–1106. doi:10.14778/1453856.1453973.

Wang, L., Khan, L. and Thuraisingham, B. (2008) 'An effective evidence theory based K-nearest neighbor (KNN) classification', 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 797–801. doi:10.1109/wiiat.2008.411.

Zhang, Z. (2016) 'Introduction to machine learning: K-Nearest Neighbors', Annals of Translational Medicine, 4(11), pp. 218–218. doi:10.21037/atm.2016.03.37.