Improvements in kNN2.java for Classification Accuracy

Feature Scaling

Feature scaling is applied to both training and test data to standardize the range of independent variables. This normalization process prevents features with larger scales from dominating the distance metric in the kNN algorithm. The following steps were taken:

Min-Max Scaling

For each feature, the minimum and maximum values across all instances were computed. The min-max scaling formula was then applied to normalize the data within the range [0, 1].

ScaledValue=Max−Min/Value−Min

Implementation

The featureScaling method iterates through each feature, calculates the min and max values, and applies min-max scaling to each instance in the dataset.

Classification with kNN

The kNN algorithm is employed for classification after feature scaling. The betterClassify method is introduced to encapsulate the improved classification process.

kNN with k=1

In this implementation, a small k value (k=1) is chosen for the kNN algorithm. A smaller k emphasizes local patterns and reduces the impact of outliers.

Classification Accuracy

The classification accuracy is computed using the computeAccuracy method, comparing predicted labels with actual labels.

Possible improvements

One specific improvement that could significantly enhance the existing code's performance is exploring alternative distance metrics in the kNN algorithm. The current implementation employs the default Euclidean distance for nearest neighbour search. However, the data could be better suited to the Manhattan distance for example. By using a different distance the kNN will be more accurate due to the change in calculating the distance.