Report on classification algorithms implementation

The classifiers was implemented in Python and evaluated on the Iris dataset.

One rule

The OneRule classifier is a simple machine learning algorithm that selects the single best feature to make predictions. It evaluates each feature individually to find the one with the lowest error rate when used for classification.

The implemented steps:

- For each feature, count how often each class occurs for each feature value.
- Calculate the error rate for each feature based on misclassifications.
- Select the feature with the lowest error rate as the rule for classification.

Features were scaled to a specified range, and the classifier identified the best feature for classification. The evaluation showed the selected feature and its associated error rate.

Naive bayes

The Naive Bayes classifier is a probabilistic machine learning algorithm based on Bayes' Theorem. It assumes that the features are independent given the class label, which simplifies the computation of the conditional probabilities.

The implemented steps:

- Calculate the prior probabilities for each class.
- Compute the likelihood of each feature given each class.
- Use Bayes' Theorem to calculate the posterior probability for each class given the feature values.
- Predict the class with the highest posterior probability.

Features and target labels were used to compute the prior probabilities and likelihoods for each class. The classifier then used these probabilities to predict the class labels for new data.

Decision tree

The Decision Tree classifier is a tree-based algorithm that splits the data into subsets based on the most significant feature at each node, creating a model that predicts the target value.

The implemented steps:

- For each feature, calculate the information gain from a potential split.
- Choose the feature with the highest information gain to make the split.
- Recursively repeat the process for each child node until a stopping condition is met (e.g., maximum depth or minimum samples per split).
- Assign the most common class label to the leaf nodes.

The classifier created a tree structure by splitting the data based on the features that provided the highest information gain.

Knn

The K-Nearest Neighbors (KNN) classifier is a simple and intuitive algorithm that classifies a data point based on the majority class of its k-nearest neighbors in the feature space.

The implemented steps:

- Determine the value of k (the number of neighbors to consider).
- Calculate the distance between the data point to be classified and all points in the training dataset.
- Identify the k-nearest neighbors based on the smallest distances.
- Assign the class label that is most common among the k-nearest neighbors.

By choosing an appropriate value of k and calculating distances, the classifier predicted the class labels based on the majority vote of the nearest neighbors.