K-NEAREST NEIGHBORS (KNN) CLASSIFICATION – REPORT

Objective

To understand and implement K-Nearest Neighbors (KNN) for classification problems using the Iris dataset. The aim is to explore how changing the value of K affects performance, visualize decision boundaries, and evaluate accuracy using metrics like confusion matrix.

Comparison Table

Step	Mini Guide Description	Implemented Code	Extra Feature
1	Choose a classification dataset and normalize features	Iris dataset used, standardized using StandardScaler	Feature names and target names extracted and reused
2	Use KNeighborsClassifier from sklearn	Trained KNN model using different values of K	Experimented with K = 1, 3, 5, 7
3	Experiment with different values of K	Model tested with multiple K values	Printed accuracy and plotted confusion matrix for each
4	Evaluate model using accuracy, confusion matrix	Used accuracy_score and confusion_matrix	Plotted heatmap of confusion matrix using seaborn
5	Visualize decision boundaries	Plotted KNN boundaries using first 2 features	Used colored contours and class-based scatterplots

Summary

The KNN classifier was successfully applied to the Iris dataset. The dataset was normalized using StandardScaler, and the model was trained and evaluated for different values of **K (1, 3, 5, 7)**.

For each K, accuracy was measured and confusion matrices were plotted, giving a clear view of how well the model classified different species of iris flowers.

Finally, the decision boundaries were plotted using only the **first 2 features** to visually show how the model separates classes in feature space. Colored contour plots and scatter plots made it easy to interpret decision regions.

Conclusion

KNN proved to be a simple but effective classifier when features are scaled properly. The best K value depends on the dataset, and visualization helps in understanding model behavior. This task provided strong hands-on experience with lazy learning and neighborhood-based classification.