# Project Report: K-Nearest Neighbors (KNN) Classification

## Objective

The primary objective of this task was to implement the K-Nearest Neighbors (KNN) algorithm on a classification problem and understand its behavior through model evaluation and visualizations. The focus was on:  
- Instance-based learning  
- Feature normalization  
- K-value experimentation  
- Visualization of decision boundaries

## Dataset Description

Dataset Used: Iris Dataset  
Source: sklearn.datasets.load\_iris  
  
Features: 4 numeric features  
- sepal length  
- sepal width  
- petal length  
- petal width  
  
Target Classes:  
- Setosa  
- Versicolor  
- Virginica

## Tools and Technologies

Programming Language: Python  
Libraries Used:  
- pandas (data manipulation)  
- scikit-learn (modeling, evaluation, PCA)  
- matplotlib, seaborn (visualization)

## Steps Performed

1. Data Loading

Used load\_iris() from sklearn.datasets to load the Iris dataset into a DataFrame for analysis.

2. Data Preprocessing

Normalization was applied using StandardScaler to scale the data to zero mean and unit variance.  
This ensures the Euclidean distance used in KNN treats all features equally.

3. Train-Test Split

Used train\_test\_split from sklearn.model\_selection to divide the dataset into training and testing sets with an 80/20 split.

4. Model Building

Implemented the KNeighborsClassifier from sklearn.neighbors.  
Trained the model for different values of K (1 to 10) and compared their accuracy.

5. Model Evaluation

Used accuracy\_score and confusion\_matrix to evaluate performance.  
Accuracy scores for various K values were printed.  
A heatmap of the confusion matrix was created using Seaborn.

6. Dimensionality Reduction (PCA)

Reduced the feature space from 4D to 2D using Principal Component Analysis (PCA) for visualization purposes.  
This allowed plotting of decision boundaries in two dimensions.

7. Visualization of Decision Boundaries

Created a mesh grid to visualize how KNN classifies data points across the 2D PCA space.  
Plotted the decision regions and overlaid actual class-labeled points for clarity.

## Results

Best Accuracy: Observed at K = 5 with the highest performance on the test set.  
  
Visualization:  
- The confusion matrix showed that the model was highly accurate in classifying Setosa.  
- Some misclassification occurred between Versicolor and Virginica, which is typical due to feature overlap.  
- The decision boundary plot provided an intuitive understanding of how KNN separates classes in feature space.

## What I Learned

- How KNN works and why it's considered a lazy learner (no training phase).  
- The critical importance of feature normalization when using distance-based algorithms.  
- How to tune the value of K to find a balance between underfitting and overfitting.  
- The usefulness of PCA in reducing feature dimensions for effective visualization.  
- Gained hands-on experience in evaluating classifiers using accuracy and confusion matrices.  
- Learned to visualize decision boundaries, which help interpret model predictions.