Lab Assignment: 6

Objective: To implement k-NN algorithm and apply on a dataset. ¶

Name: Aakash Verma

Reg. No.: 24-08-26

Course: M.Tech.(Cyber Security)

In [3]:
```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:
```
# Load the Iris dataset
iris_data = load_iris()
features = iris_data.data
labels = iris_data.target
```

In [5]:
```
# Convert to DataFrame for easier handling
iris_df = pd.DataFrame(data=np.c_[features, labels], columns=iris_data.feature_names + ['label'

# Split the dataset into training and testing sets
train_features, test_features, train_labels, test_labels = train_test_split(features, labels, t
```

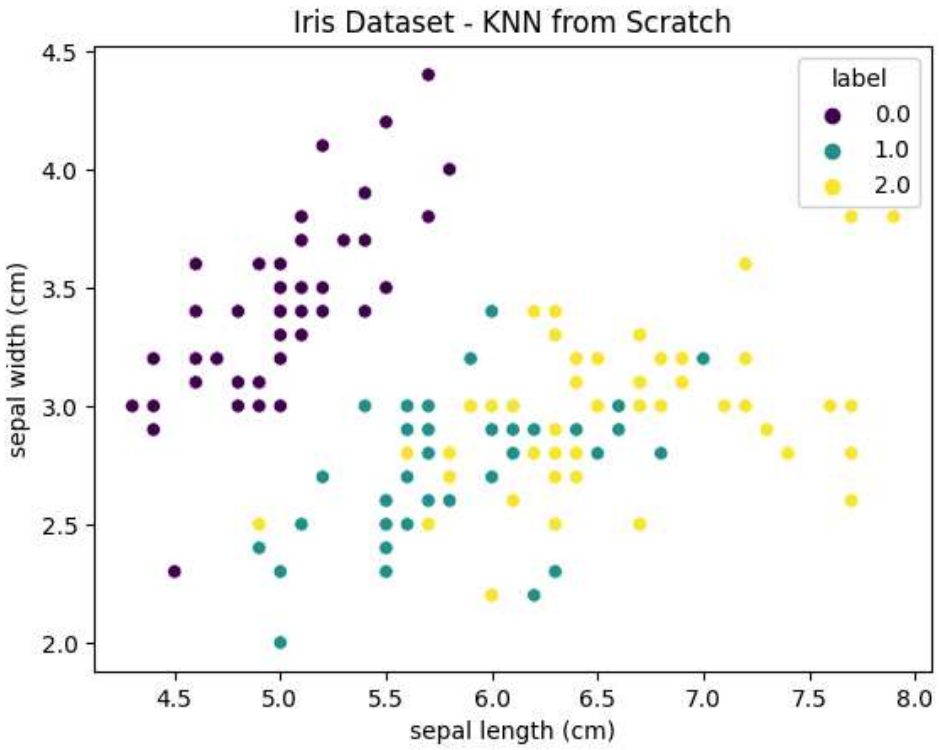
In [6]:
```
# Function to calculate Euclidean distance
def euclidean_distance(point1, point2):
    return np.sqrt(np.sum((point1 - point2) ** 2))

# KNN algorithm
def knn(train_features, train_labels, test_point, k):
    distances = []
    for i in range(len(train_features)):
        distance = euclidean_distance(test_point, train_features[i])
        distances.append((distance, train_labels[i]))
    distances.sort(key=lambda x: x[0])

    # Get the nearest k neighbors
    neighbors = [distances[i][1] for i in range(k)]

    # Majority voting
    prediction = np.argmax(np.bincount(neighbors))
    return prediction
```

In [7]:
```
# Make predictions on the test set
k = 3
predictions = [knn(train_features, train_labels, test_point, k) for test_point in test_features
```

In [8]:
```
# Calculate accuracy
accuracy = np.sum(predictions == test_labels) / len(test_labels)
print(f"Accuracy: {accuracy:.2f}")
```

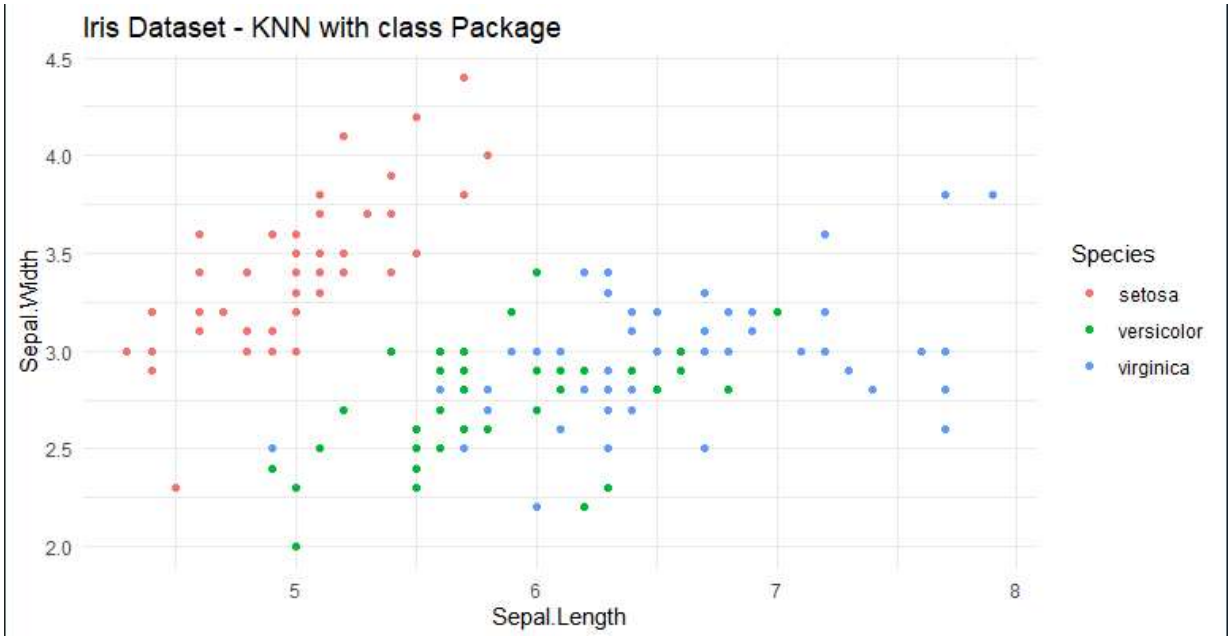
```
Accuracy: 1.00
```

In [9]:
```
# Plotting the results
sns.scatterplot(data=iris_df, x='sepal length (cm)', y='sepal width (cm)', hue='label', palette
plt.title("Iris Dataset - KNN from Scratch")
plt.show()
```



R code

In [ ]:
```
# Load necessary libraries
library(class)
library(ggplot2)

# Load the iris dataset
data(iris)

# Set seed for reproducibility
set.seed(42)

# Split the dataset into training and testing sets
train_indices <- sample(1:nrow(iris), size = 0.8 * nrow(iris))
train_data <- iris[train_indices, ]
test_data <- iris[-train_indices, ]

# KNN classification
k <- 3  # Number of neighbors
predicted_labels <- knn(train = train_data[, -5],
                        test = test_data[, -5],
                        cl = train_data$Species,
                        k = k)

# Calculate accuracy
accuracy <- sum(predicted_labels == test_data$Species) / nrow(test_data)
cat("Accuracy:", round(accuracy, 2), "\n")

# Plotting the results
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +
  labs(title = "Iris Dataset - KNN with class Package") +
  theme_minimal()
```



Conclusion:

High Accuracy: The model achieved an accuracy of 100% on the test set, suggesting either perfect classification or potential overfitting due to the simplicity and size of the dataset.

Performance Assessment: While the high accuracy is promising, further validation methods, such as k-fold cross-validation, are recommended to ensure the model's robustness and generalizability.