Lab Assignment: 5

Objective: To implement KNN and apply on iris dataset.

Name: Aakash Verma

Reg. No.: 24-08-26

Course: M.Tech.(Cyber Security)

In [3]:
```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [4]:
```
class KNN:
    def __init__(self, k=3):
        self.k = k

    def fit(self, X, y):
        self.X_train = X
        self.y_train = y

    def predict(self, X):
        return np.array([self._predict_single(x) for x in X])

    def _predict_single(self, x):
        distances = np.linalg.norm(self.X_train - x, axis=1)
        k_indices = np.argsort(distances)[:self.k]
        k_nearest_labels = [self.y_train[i] for i in k_indices]
        return np.bincount(k_nearest_labels).argmax()
```

In [5]:
```
# Load Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
```

In [6]:
```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4
```

In [7]:
```
# Initialize and fit the KNN classifier
knn = KNN(k=3)
knn.fit(X_train, y_train)
```

In [8]:
```
# Make predictions
predictions = knn.predict(X_test)
```

In [9]:
```
# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 1.00
```

Conclusion:

1. NN effectively classified Iris species, demonstrating high accuracy through distance metrics.

2. The implementation showcased KNN's simplicity, making it an accessible introduction to machine learning concepts.

In [ ]: