Implementation of KNN from Scratch

Name: Ankita Dasgupta

Roll No: J014

In [1]:
```
import numpy as np
import pandas as pd
from scipy.stats import mode
from numpy.random import randint
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [4]:
```
X = load_iris().data
y = load_iris().target
cols = load_iris().data
```

In [5]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[5]: ((120, 4), (30, 4), (120,), (30,))

In [6]:
```
# Euclidean Distance Function
def e_d(p1,p2):
    return np.sqrt(np.sum((p1-p2)**2))
```

In [9]:
```
def kNN(x, y, inp, k):
    labels = []
    for i in inp:
        dist = []
        for j in range(len(x)):
            d = e_d(np.array(x[j,:]), i)
            dist.append(d)
        dist = np.array(dist)
        dist_new = np.argsort(dist)[:k]
        l = y[dist_new]
        # Majority Voting
        lab = mode(l)
        lab = lab.mode[0]
```

```
        labels.append(lab)
    return labels
```

In [13]:
```
from sklearn.neighbors import KNeighborsClassifier as KNN

clf = KNN(n_neighbors=5)
clf.fit(X_train, y_train)
```

Out[13]:
```
KNeighborsClassifier()
```

In [14]:
```
y_pred_1 = kNN(X_train, y_train, X_test, 5)
y_pred_2 = clf.predict(X_test)
print("Our Model", accuracy_score(y_test,y_pred_1))
print("sklearn Model", accuracy_score(y_test,y_pred_2))
```

```
Our Model 0.9666666666666667
sklearn Model 0.9666666666666667
```