Experiment 10: KNN from scratch

Name: Ashish Nanda

Roll No.: J041

In [1]:

```
import numpy as np
from scipy.stats import mode
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [2]:

```
class KNN:

    def __init__(self, n_neighbors=5):
        self.n_neighbors = n_neighbors

    def euclidean(self, d1, d2):
        distance = np.sqrt(np.sum((d1 - d2) ** 2))

        return distance

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

    def predict(self, X_test):
        y_pred = np.zeros(len(X_test))

        for indx, X in enumerate(X_test):
            distance_and_neighbors = []

            for X_tr, y_tr in zip(self.X_train, self.y_train):
                dist = self.euclidean(X, X_tr)
                distance_and_neighbors.append((dist, y_tr))

            distance_and_neighbors = sorted(distance_and_neighbors, key=lambda k
: k[0])
            neighbors = [n[1] for n in distance_and_neighbors[:self.n_neighbors
]]
            y_pred[indx] = mode(neighbors)[0][0]

        return y_pred
```

Testing the model on Iris dataset

In [3]:

```
iris_df = load_iris()
```

In [4]:

```
X = iris_df.data
y = iris_df.target
```

In [5]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
state=22, stratify=y, shuffle=True)
```

In [6]:

```
model = KNN(n_neighbors=7)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

In [7]:

```
score = accuracy_score(y_test, y_pred)
score
```

Out[7]:

```
0.9666666666666667
```