Week 4 - k Nearest Neighbours

MACHINE INTELLIGENCE LABORATORY

Teaching Assistants
vighneshkamath43@gmail.com
sarasharish2000@gmail.com
abaksy@gmail.com

k-Nearest Neighbours is one of the most famous supervised classification and regression algorithms. The beauty of the algorithm is that we can observe it in our real life, i.e., our nature is to be close with people who are the most like us.

In this assignment you are required to prepare a Python class KNN which can be used by anyone for **classification**. The detailed instructions are given to you in this document.

Your task is to complete the code for the class KNN and its methods

You are provided with the following files:

1. **week4.py**
2. **SampleTest.py**

Note: These sample test cases are just for your reference.

SAMPLE TEST CASE Data

The training data is given in the form below (N=10, D=2)

	x1	x2	y
0	2.781084	2.550537	0.0
1	1.465489	2.362125	0.0
2	3.396562	4.400294	0.0
3	1.388070	1.850220	0.0
4	3.064072	3.005306	0.0
5	7.627531	2.759262	1.0
6	5.332441	2.088627	1.0
7	6.922597	1.771064	1.0
8	8.675419	-0.242069	1.0
9	7.673756	3.508563	1.0

The dataset here consists of N points each of D dimensions and the class that the data points belong to (x1 and x2 represent the dimensions, and y represents the class/category that the point belongs to). All dimensions are **numerical** and **continuous** in nature; hence you do not need to encode the data in any way.

Note that there may be more than 2 classes. Do not assume that number of classes will be 2 always!

Important Points:

1. Please do not make changes to the function definitions that are provided to you. Use the skeleton as it has been given. Also do not make changes to the sample test file provided to you. Run it as it is.
2. You are free to write any helper functions that can be called in any of these predefined functions given to you. Helper functions must be only in the file named 'YOUR_SRN.py'.
3. **Note that the code is for classification only.**
4. Your code will be auto evaluated by our testing script and our dataset and test cases will not be revealed. Please ensure you take care of all edge cases!
5. **The experiment is subject to zero tolerance for plagiarism. Your code will be tested for plagiarism against every code of all the sections and if found plagiarized both the receiver and provider will get zero marks without any scope for explanation.**
6. **Kindly do not change variable names or use any other techniques to escape from plagiarism, as the plagiarism checker is able to catch such plagiarism**
7. Hidden test cases will not be revealed post evaluation.

8. The distance metric used in this implementation is the Minkowski Distance (the parameter ‘p’ of the Minkowski Distance will be supplied as input to the KNN class constructor).
9. In case of any ambiguities during choosing the class from the K-nearest neighbours, choose the class which is **lexicographically smallest.**
10. All the classes (targets for classification) will be integer-valued
11. In case the ‘weighted’ flag is set to True, you are required to use the **reciprocal** of the distance between the point and the query as the weight of that point. Mathematically:

$$W_{x_i} = \frac{1}{d(x_i, q)}$$

week4 .py

- ✓ You are provided with structure of class KNN.
- ✓ The class KNN contains one constructor and five methods.
- ✓ Your task is to write code for the four of these five methods.

- def__init_(self, k_neigh, weighted=False, p=2)
  - This is the constructor of the class KNN
  - You are not required to edit this method. However, it is necessary to understand the initialization to implement other methods.

    parameters: k_neigh: int , weighted= Boolean , p= int/float

    k_neigh: defines the k value for the algorithm
    weighted: **if True, weighted kNN algorithm should be used else vanilla kNN should be used**
    p :defines the parameter of Minkowski distance (when p==2 ,Euclidean)

- def fit(self, data, target)
  - This method has already been implemented, and you are not required to make any changes to this function.
  - parameters: data: M x D Matrix(M data points with D attributes each)(float)
    target: M Vector(Class target for all the data points as int.)
  - Returns : The object itself

- def find_distance(self, x):
  - This method is used to find the Minkowski distance between all pairs of points in the train dataset
  - parameter: N x D Matrix(N points in the train dataset with D attributes each)(float)
  - Returns: Distance between each input to each data point in the train dataset
    (N x N) Matrix (N points in the train dataset)(float)
- def k_neighbours (self, x):
  - Find the K points that are closest to each of the points in the train dataset (i.e., find the k Nearest Neighbours of every point in the train dataset.
  - Args:
    x: N x D Matrix(N inputs with D attributes each)(float)
  - Returns:
    A list consisting of: [neigh_dists, idx_of_neigh]
    neigh_dists -> N x k Matrix(float) - Distance of all input points to their k closest neighbours
    idx_of_neigh -> N x k Matrix(int) - The row index in the dataset of the k closest neighbours of each input
- def predict(self, x):
  - This method is used to predict the target value of the inputs.
  - Parameter: x: N x D Matrix(N inputs with D attributes each)(float)
  - Returns: pred: N Vector(Predicted target value for each input)(int)

- def evaluate(self, x, y):
  - This method is used to evaluate Model on test data using classification: accuracy metric
  - Parameter: x: Test data (N x D) matrix(float) y: True target of test data(int)
  - Returns: accuracy : (float.)

1. **You may write your own helper functions if needed**
2. **You can import libraries that come built-in with python 3.7**
3. **You cannot change the skeleton of the code**
4. **Note that the target value is an int**

SampleTest.py

1. This will help you check your code.
2. Passing the cases in this does not ensure full marks, you will need to take care of edge cases
3. Name your code file as YOUR_SRN.py
4. Run the command

python3 SampleTest.py --SRN YOUR_SRN

if import error occurs due to any libraries that is mentioned in the skeleton code try:

```
python3.7 SampleTest.py --SRN YOUR_SRN
```