CSE 6363 - Machine Learning   Spring 2024
Due Date: Sep 21, 2024, 11:59 PM

PLEASE READ ALL PARTS OF THE DOCUMENTS CAREFULLY

Assignment Algorithm: K Nearest Neighbor
Dataset: P1input2024.txt. The file is tab separated.

- The dataset is comprised of X, Y coordinates of a parking lot along with a label 1 or 0.
- When a car is parked in a spot, the label says 1 and displays a label 0 otherwise.
- There are several trees in the parking lot.
- Each tree takes up a full parking spot and a vehicle cannot be parked in the spot used up by a tree.
- There are 4 trees in the parking lot: T1 (4, 0), T2 (0, 4), T3(3, 9), and T4(7, 9)

Q1 points: 20   **Q2** points: 80

Q1 Consider the problem where you are given the occupancy information of a subset of the parking spots and you want to use **KNN to** predict whether any other parking spots are occupied or not. **You can use any libraries in Q1.**

A. Use 3 different values for K (3,5,7) and Euclidean distance to measure the distance between data points. Split the dataset into 80% train and 20% test datasets.

B. Run your experiments for the following:

a. Values of K: 3, 5, and 7

b. For each of the K values, print the following outputs **on the screen and in a file.**

i. Accuracy
ii. Recall
iii. Precision
iv. F1 score
v. Confusion matrix

c. Display Accuracy, Recall, Precision, F1 score, and the confusion matrix for each K (3,5,7) as:

i. outputs on the screen and
ii. Store all outputs in the file **P1Output2024.txt**

d. In your code, screen, and file output, clearly label the Question numbers (with or without the libraries), subpart numbers, values of K, and the corresponding values of Accuracy, Recall, Precision, F1-score, and Confusion Matrix for **Q1.**

Q2 Consider the problem where you are given the occupancy information of a subset of the parking spots and you want to use **KNN to** predict whether any other parking spots are occupied or not. **You can NOT use any libraries except for NumPy and Math. Every function needs to be implemented from scratch.**

A. Add two additional columns to your dataset. The first column to be added is the Manhattan distance between the **parking spot and the closest tree.** Use the parking spot's Manhattan distance from all trees to find the distance from the closest tree. The second column to be included is the name of the **closest tree:** T1, T2, T3, or T4. In other words, if a parking spot is at the shortest distance of 3 from the tree T4, then **3, T4** needs to be included in the record. Then create one hot encoding for the tree names. **Store these long records in a new file called P1input2024LongRecords.txt This needs to be a tab separated file. Use this newly generated file P1input2024LongRecords.txt for all experiments in Q2.**

B. Use 3 different values for K (3,5,7) and Euclidean distance to measure the distance between points. Split the dataset into 80% train and 20% test dataset.

C. Run your experiments for the following:

a. Values of K: 3, 5, and 7 and input feature vector **set 1**, **set 2,** and **set 3** where

i. Input feature vector **set 1** includes X and Y of the parking spot.
ii. Input feature vector **set 2** includes X, Y, Manhattan distance from the closest tree of the parking spot.
iii. Input feature vector **set 3** includes X, Y, Manhattan distance from the closest tree of the parking spot, and the one hot encoding for the tree name.

b. For each of the combinations in Q2.C.a, print the following outputs on the **screen** and in **the file**.

i. Accuracy

        ii. Recall
        iii. Precision
        iv. F1 score
        v. Confusion matrix

    c. Display Accuracy, Recall, Precision, F1 score, and the confusion matrix for each K (3,5,7) and sets 1, 2, and 3 as:
        i. outputs on the screen and
        ii. **Append** all outputs to the file **P1Output2024.txt**
    d. In your code, screen, and file output, **clearly label the Question** numbers (with or without the libraries), **subpart numbers, values of K, type of input set (1,2, and 3)** and the corresponding values of Accuracy, Recall, Precision, F1-score, and Confusion Matrix for **Q2.**

Documentation required:

- Submit the clean data files that you generated while executing Q1 and Q2. These are:
    - **P1input2024LongRecords.txt**
    - **P1Output2024.txt**
- Submit a **README file that** explains the following:
    - How to run the code including any environment dependency (language versions).
    - Honor code (stated in red in this document)
- Implement your code yourself without copying from anywhere. **Cite any sources that you may have used.**
- **The code will be tested for plagiarism. Any plagiarism will result in zero for all students involved. Submission of a python file (.py) for the plagiarism test is necessary. Failing to submit a .py file will result in marks deduction (-50 points).**
- **The program should not take more than 5 minutes to run. If it takes longer than 5 minutes to show the required output, marks will be deducted.**

Some rules to follow:

1. Make Sure that the TA (Teaching Assistant) can run your code. TAs may ask you to demo your project. **The student not present for the requisite demo will receive a zero.**
2. Ensure that your code is readable (indented and commented).
3. **Project will be completed by and submitted by individuals**.
4. All files need to be zipped in a zip file that is named as follows:
    a. CSE6363-sectionNo-P1-1000XXXXX (individual work)
5. Handwrite, sign, and date (with date of submission) a copy of the Honor Code (shown below) and share the image as part of your project; a handwritten, signed, and dated (with the date of submission) copy of the Honor Code must be included with every project and exam submission. (Failing to include will cost 20 points)
6. **Students must NOT share the project and their solutions even after the semester is over or after their graduation.** However, they can show their projects during their interviews to the interviewer. They are also required to not discuss the solution with others or use anyone else's solution. **Any violation of the policy will result in a 0 for all students concerned.**

HONOR CODE

I pledge, in my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code

I will not participate in any form of cheating/sharing the questions/solutions.