Machine Learning [Fall 2022]
Instructor: Dr. Hanaa Mobarez

Faculty of Computers and AI
Cairo University

Assignment 2
Decision Trees and K-nn

Dataset Description:

Whenever you go to the bank to deposit some cash money, the cashier places banknotes in a machine that tells whether a banknote is real or not. In the "BankNote_Authentication.csv" you have four features: variance, skew, curtosis and entropy and the class attribute refers to whether or not the banknote is real or forged.

Problem 1 [Decision Trees using Scikit-learn]:

Use the Banknote Authentication data attached with the assignment to implement the following requirements:

1. Experiment with a fixed train_test split ratio:Use 25% of the samples for training and the rest for testing.
    a. Rerun this experiment five times and notice the impact of different random splits of the data into training and test sets.
    b. Report the sizes and accuracies of these trees in each experiment.

2. Experiment with a range of train_test split ratio: Measure the impact of training set size on the accuracy and the size of the learned tree. Consider training set sizes in the range [30% - 70%] *(Start with training data size 30% , 40% .... Until you reach 70%)* and for each training set_size :
    a. run the experiment with five different random seeds.
    b. calculate mean, maximum and minimum accuracy at each training set_size.
    c. measure the mean, max and min tree size.
    d. store your statistics in a report.
    e. Draw two plots: 1) shows accuracy against training set size and 2) the number of nodes in the final tree against training set size.

Problem 2 [KNN]:

Use the Banknote Authentication data to implement your own simple KNN classifier using python, (Don't use any built-in functions), divide your data into 70% for training and 30% for testing.

1. **If there is a tie in the class predicted by the k -nearest neighbors, then among the classes that have the same number of votes, the tie should be broken in favor of the class that comes first in the Train file.**
2. **Each feature column should be normalized separately from all other features. Specifically, for both training and test objects, each feature should be transformed using the function: *f(v) = (v - mean) / std*, using the mean and std of the values of that feature column on the TRAINING data.**
3. **Use Euclidean distance to compute distances between instances.**
4. **Experiment with different values of k=1,2,3….9 and output the following:**
    a. **The value of k used for the test set on the first line followed by summary info for the current k value: 1) number of correctly classified test instances, and the total number of instances in the test set and accuracy.**

Output Example:

k value : 3
Number of correctly classified instances : 238 Total number of instances : 445
Accuracy : 0.5348314606741573

SUBMISSION RULES

1. **Deadline: 20 Dec. 2022 10:00 PM**
2. **Students should be from the same lab or from labs given by the same TAs.**
3. **Deliver your code as ".py" files not notebook extensions.**
4. **Name your zipped folder that contains [Code + Report] as follows: ID1_ID2_ID3…_G# (i.e. 2020111_2020333_CS1)**
5. **Team members are Min: 3 and Max: 4**
6. **ONLY one member should upload the zipped folder to avoid multiple submissions for the same team.**
7. **Penalty will be applied in case you violated the previous rules.**

GRADING CRITERIA

Problem 1 [DT]	[3.0]
Experiment with Fixed Training Set size	0.5
Experiment with a range of train_test split ratio	1.5
Report for both experiments including the two plots in experiment 2.	1.0
Problem 2 [Knn]	**[3.0]**
Knn algorithm (+ handling tie case)	1
Normalization + Euclidean distance Calc.	1
Experiment with different k values and print summary info as required	1