Assignment 2

Due on December 5, 2022 (23:59:59)

Instructions. The goal of this problem set is to make you understand and familiarize yourself with the decision tree algorithm. You will experiment with the decision tree model (by using the ID3 algorithm) on the Flower Species dataset.

PART I: Flower Species Classification

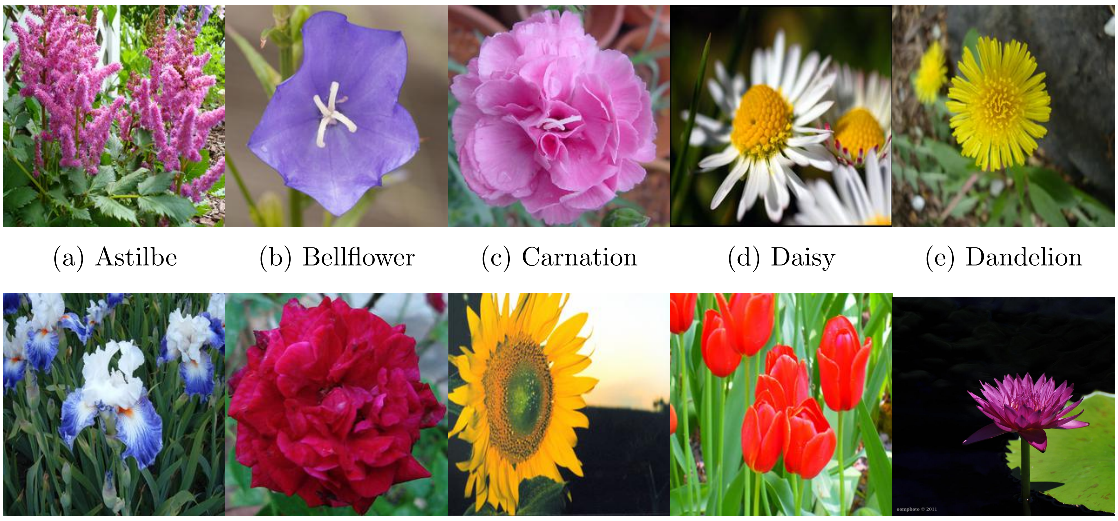
In this assignment, you will implement a decision tree model to classify a flower species.

A dataset [1] is provided for your training phase. You will use the given train, test, and validation split. You will implement the ID3 concept within the scope of your decision tree model. You will implement ID3 for discrete attributes on the dataset.

The Flower Species Dataset

The Flower Species Dataset is an image dataset that consists of 10 different flower species.

- You can download the dataset from given link
- Dataset includes train and test images. Each species dataset includes 600 images for training, 50 images for validation, and 50 images for testing.



(a) Astilbe (b) Bellflower (c) Carnation (d) Daisy (e) Dandelion

(f) Iris (g) Rose (h) Sunflower (i) Tulip (j) Water_lily

Figure 1: Examples for flower species.

Features

You need to use the features that you think are proper for your building detection assignment. For these implementations, you can use the CV library (tutorial). Some features are listed below:

Image Size: You can resize images to a small size for a short training time.

Color space: You can change the color space of images (RGB, HSV, and Grayscale).

Shape features: Shape is an important and powerful feature for image processes. You can use shape information extracted using a histogram of edge detection. Edge information in the image is obtained by using the Canny Edge Detector, Sobel Filter, and Laplacian Filter.

Texture features: The texture feature is extracted usually using a filter-based method. The Gabor filter is a frequently used filter in texture extraction.

Also, you can apply smoothing, sharpening, and noise-removing methods. You can use more than one feature by concatenating them.

Classfication Performance Metric

You will compute ”Accuracy”, ”Precision”, ”Recall” and ”F1 Score” of your model to measure the success of your classification method based on your constructed confusion matrix, in which TP means ”True Positive”, TN means ”True Negative”, FP means ”False Positive” and finally FN means ”False Negative”:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\textbf{F1 Score} = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \tag{4}$$

You will report these four measurement metrics test set and finally, write the rules for your best decision tree model variation with respect to your dataset. While writing a decision tree model’s rules, you must print all root-to-leaf paths in left-to-right order.

Error Analysis for Classification

- Find a few misclassified images and comment on why you think they were hard to classify.
- Compare the performance of different ID3 model variation choices for your dataset. Wherever relevant, feel free to discuss computation time in addition to the classification rate.

Steps to Follow for Classification

1. Read your classification data. Training and test split given on your dataset.
2. Extract features for each image in the training set (image size, Grayscale, Gabor, etc.).
3. Train your ID3 decision tree model with respect to your features.
4. For each given test sample measure the features' quality.
5. Compute and report your "Accuracy", "Precision", "Recall" and "F1 Score" of your different ID3 model parameters. Finally, write the rules of your best-performing decision tree model with respect to these four metrics mentioned in "Classification Performance Metric".
6. Report your findings in the "Error Analysis for Classification" section.

PART II: Pruning Decision Tree

You are also expected to prevent overfitting on decision tree by pruning the twigs of the tree. The twigs are the nodes whose children are all leaves. For the pruning process, you will use the given train, test, and validation split of the dataset. The algorithm you follow for the pruning process is below:

- Create a "Last Accuracy" variable and set this accuracy to the accuracy of your decision tree model on the validation set before the pruning process.
  - Step 1: Catalog all twigs in the tree
  - Step 2: Find the twig with the least Information Gain
  - Step 3: Remove all child nodes of the twig
  - Step 4: Relabel the twig as a leaf (Set the majority of "Positive" or "Negative" as leaf value)
  - Step 5: Measure the accuracy value of your decision tree model with removed twig on the validation set ("Current Accuracy")

  - If "Current Accuracy $\geq$ Last Accuracy" : Jump to "Step1"

    Else : Revert the last changes done in Step 3,4 and then terminate

After the pruning process, you must write the rules and accuracy values on the test set for both your pre-pruning decision tree and post-pruning decision tree. Also, you must compare them and state in your report the differences between these two models. And, report after the pruning process, which redundant features/attributes are pruned, and comment about why you think that features are pruned.

Implementation Details

- You can't use ready-made libraries for your decision tree implementation, pruning process implementation, and discretization process implementation. You must implement these on your own.
- You can use ready-made libraries for computing "Accuracy", "Precision", "Recall" and "F1 Score" metrics and for creating your confusion matrix.
- You may use Numpy array functions for your intermediate implementation steps for your implementations
- You may use "Pandas" library.

Submit

You are required to submit all your code in a Jupyter Notebook, along with a report in ipynb format, which should also be prepared using Jupyter Notebook. The code you submit should be thoroughly commented on. Your report should be self-contained and include a concise overview of the problem and the details of your implemented solution. Feel free to include pseudocode or figures to highlight or clarify specific aspects of your solution. Finally, prepare a ZIP file named name-surname-a2.zip containing:

- assignment_2.ipynb (including your report and code)
- assignment_2.py (py file version of your ipynb file)
- Do not send the dataset.

The ZIP file will be submitted via Google Classroom. Click here to accept your Assignment 2.

Grading

- Code part: ID3: 30 points, Pruning: 20 points, other codes parts (well-written): 10 points
- Theory part(40 points): Analysis of the results for Decision Tree Classfication and Pruning

Note: Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects on the results. You can create a table to report your results.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.