CMPE 462 - Spring 2021
Assignment 3

Introduction

This assignment consists of 2 parts.
The first one is about decision trees and the second part is about SVM.

Part1

In Part 1 of this assignment, you will implement binary **Decision Tree** (DT) from scratch. Each node in your DT can have 2 child nodes. Each decision node can check one parameter only.

The dataset is the famous Iris Dataset ([1]). The dataset is given to you in csv format with header. The class label is in the last column. You will be using only two classes: Iris-setosa and Iris-virginica. There are 50 samples for each class. Use first 40 as train set and last 10 as test set for the two classes. (total of 80 as train set and 20 as test set)

The steps of Part1 are below:

- Step1: Implement DT with information gain and apply on the dataset.
- Step2: Implement DT with gain ratio and apply on the dataset.

Your program will output the name of the root parameter of the tree and the test accuracy (space separated) as below:

```
DT sepal-length 0.70
```

In your assignment report, include plot of resulting trees. As there are only 4 features, you can choose to plot manually. Calculate and report accuracy for test set.

[1] https://archive.ics.uci.edu/ml/datasets/iris

Part2

In Part 2 of this assignment, you will focus on **Support Vector Machines** (SVM). You can use LibSVM library([2],[3]) for Python.

The dataset is the Breast Cancer Wisconsin dataset from UCI ([4]). The dataset is given to you in csv format with header. There are 569 samples in two classes. Use first 400 as train set and the rest as test set. You will not use the first column which is labelled as "id". The class label is in the second column.

The steps of Part2 are below:

- Step1: Apply SVM with 5 different C values for a fixed kernel. Report accuracy and number of support vectors.
- Step2: Apply SVM with different kernels for a fixed C value. Report accuracy and number of support vectors.

For X as kernel and Y as C value, your program achieves A accuracy for test set and finds S (number of) support vectors. Then the output of your script will be (no extra log prints) as below. Values are space separated. There are 5 items. No space before or after =, no space in kernel name.

```
SVM kernel=X C=Y acc=A n=S
```

As you will try several settings, your script will output several lines.

```
SVM kernel=X1 C=Y1 acc=A1 n=S1
SVM kernel=X2 C=Y1 acc=A2 n=S2
SVM kernel=X3 C=Y1 acc=A3 n=S3
```

In your report, present and discuss changes in test accuracy for Step1 and Step2. Discuss the changes in number of support vectors. Try to reason on these results.

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvm/
[3] https://pypi.org/project/libsvm/
[4] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

Base Environment

You will be implementing your code with Python 3.6.
You need to create a python virtual environment with Anaconda for your project. After installing Anaconda, a base environment can be created with below commands:

```
conda create -n 462assignment python=3.6
conda activate 462assignment
```

While you keep working on your models, you will need to import additional libraries. List these libraries in a requirements.txt file. State any special versions if needed. A sample requirements file can be as below:

```
scikit-learn >= 0.22.2
scipy
pandas
sentencepiece==0.1.91
```

For grading, we will load your requirements with the command below:

```
python3 -m pip install -r requirements.txt
```

Before submission, test your code on a clear new conda environment by installing additional libraries from your requirements file. Because, there will be penalty if your code doesn't run like this.

Grading Details

The assignment will be graded over 100 points. You will be graded for your code and report.

- 20 points for report
- 40 points for code (Part1)
  - 20 points for step 1
  - 20 points for step 2
- 40 points for code (Part2)
  - 20 points for step 1
  - 20 points for step 2

We will run your code on a clear new conda environment. First we will load your requirements.txt file. Then we will test your code with below commands:

- Part1

  ```
  python3 assignment3.py part1 step1
  python3 assignment3.py part1 step2
  ```

  Consider second command, you will run DT with gain ratio.

- Part2

```
python3 assignment3.py part2 step1
python3 assignment3.py part2 step2
```

Consider second command, you will run SVM with different C values.

Submission Details

This is an individual assignment. Your code should be original. Any similarity between submitted assignments or to a source from the web will be accepted as cheating.

If you have any further questions, send an e-mail to the course page on Piazza.

- The deadline for submitting Assignment 3 is **June 1, 2021 - 23:59**.
- There will be 2 submissions open for this assignment.
- Submission 1:
  - You should submit 3 items:
    - your Python script, assignment3.py
    - your requirements.txt file, blank file if no additional library is needed
    - your assigment report in pdf, 462_assignment3_<studentid>_report.pdf, example 462_assignment3_20181123456_report.pdf
  - You should compress all submission items in a zip file with name as 462_assignment3_<studentid>.zip, example 462_assignment3_20181123456.zip
  - The zip will be submitted on Moodle.
- Submission 2:
  - You should also submit your reports in Turnitin submission on Moodle.