Homework Programming Assignment 1: Decision Tree Classifier

Due Time: September 23, 2019, 3:00PM

Contents:

You are given data generated from three blackboxes: blackbox11, blackbox12, and blackbox13. The description and tasks for each blackbox are the same. The following instruction is for blackbox11 as an example, and you can apply the same to the blackbox12 and blackbox13 as well.

Task Description:

Please use ***Python 3*** to implement your homework assignment. In this assignment, you are given a set of training data and a set of testing data generated from the same blackbox, say blackbox11, which you may not know the secret function inside.

For blackbox11, you are given:

- `blackbox11_train.csv`: labeled training data generated from a blackbox11
- `blackbox11_test.csv` : unlabeled testing data
- `blackbox11_example_predictions.csv`: example output, which is also the true class labels for `blackbox11_test.csv` so you can get to know the format of output and measure your decision tree's performance while you are developing your program.

Your task is to implement a decision tree learner, named as `decisiontree.py`, that will

(1) construct a decision tree classifier from the given training data,
(2) use the learned tree classifier to classify the unlabeled test data, and
(3) output the predictions of your classifier on the test data into a file named `blackbox1*_predictions.csv` in the ***same*** directory as the `.py`

Your program will take two input files and produce one output file as follows:

```
python3 decisiontree.py training_data_path testing_data_path
⇒ prediction_file
```

For example,

```
python3 decisiontree.py blackbox11_train.csv blackbox11_test.csv
⇒ blackbox11_predictions.csv
```

Note: input files may not be in the same directory as your python script [1].

In other words, your algorithm file `decisiontree.py` will take ***labeled training data***, ***unlabeled testing data*** as input, and your classification predictions on testing data as output.

[1] `os.path.basename(`*`path`*`)` can be used to extract the base name of pathname *path*.

In your implementation, ***please do not use any existing machine learning library call***. You must implement the algorithm yourself. Please develop your code yourself and do not copy from other students or from the Internet.

The format of `*_train.csv` looks like:

x1, x2, y

Where x1 and x2 are the attribute values and y is the label, and `*_test.csv` are unlabeled.

Your output `blackbox1*_predictions.csv` will look like

1
0
0
… (A single column indicates the predicted labels for each unlabeled sample in the input test file)

The format of your `blackbox1*_predictions.csv` file is crucial. It has to be in the ***exact same name and format*** so that it can be parsed correctly to compare with true labels by grading scripts.

When we grade your algorithm, we will use the same training data but some unlabeled ***hidden*** testing data (generated from the same blackbox11) instead of the testing data that was given to you. Your code will be autograded for technical correctness. Please name your file correctly, or you will wreak havoc on the autograder.

Your submission:

- Submit your `decisiontree.py` to ***Vocareum.*** See *Appendix* for more information.
- Create a single **`.zip`** (`yourUSCId.zip`) which contains:
  - `decisiontree.py`
  - `report.pdf`, a ***brief*** report indicates your ***training accuracy, testing accuracy*** and ***decision tree diagram*** for blackbox11, blackbox12, and blackbox13***.***

  Submit your zipped file to the blackboard.

Rubric:

10 points in total

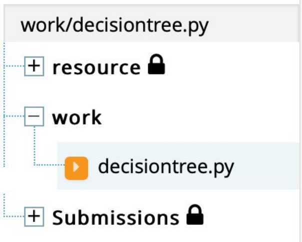
- Program correctness(6 points): your program always works correctly and meets the specifications
- Documentation(2 points): your code is well commented and the submitted tree looks reasonable
- Performance (2 points):
  - overall good performance on reported train and test accuracy (1 point)
  - accuracy on hidden testing data (1 point)

Appendix

Homework must be submitted through Vocareum. Please only upload your code to the `/work` directory. Don't create any subfolder or upload any other files. Please refer to http://help.vocareum.com/article/30-gettingstarted-students to get started with Vocareum.

How to submit

1. Log in to Vocareum website and you should see ***Assignment 1*** under INF 552 Course
2. Click ***My Work***
3. Select ***work*** folder on the left
4. You can either create a new file or upload files from your local machine. Make sure to name your python file correctly based on homework description



5. Click on ***Submit*** and press ***Yes.*** You are able to submit as many times as you want

You can test your code on your local machine in the way described in the homework handout. On vocareum, you only need to ***Submit,*** then the "***Passed/Failed***" message will appear automatically. You can check your submission results in the Terminal or under Details tab. If your code works correctly, you should see the results as below:

```
Submission Report
[Executed at: Sun Sep 8 18:32:08 PDT 2019]
==================================================
blackbox11 passed, test accuracy: 90.0
==================================================
==================================================
blackbox12 passed, test accuracy: 90.0
==================================================
==================================================
blackbox13 passed, test accuracy: 90.0
==================================================
```

Here, test accuracy is based on the test data given to you, so you can compare this with the results on your local machine to make sure everything works properly on Vocareum.

After the deadline, your ***final submission*** will be graded using hidden test data, and you will be able to see your grades on Vocareum.