TCSS 555: Data Mining/Machine Learning
Spring 2017, Homework #2
Due: Monday, Nov 6

Please submit a hard copy of your answers to the homework problems below. Staple all of your pages together (and order them according to the order of the problems below) and have your name on each page, just in case the pages get separated. Write legibly (or type) and organize your answers in a way that is easy to read. Neatness counts!

For each problem, make sure you have acknowledged all persons with whom you worked. Even though you are encouraged to work together on problems, the work you turn in is expected to be your own. When in doubt, invoke the Gilligan's Island rule (see the syllabus) or ask the instructor.

All homeworks are due at the beginning of the lecture on the due date. I will accept one homework up to one lecture late without penalty. You do not need to inform me – I will accept it automatically, no questions asked or documentation required.

---

1. **Nearest neighbor.** (2pts) You are given 6 training examples for a binary classification problem as follows:

X_1	X_2	Y
10	0	$+$
0	-10	$+$
5	-2	$+$
5	10	$-$
0	5	$-$
5	5	$-$

Plot these points in a 2-dimensional grid and show the decision boundary resulting from 1-Nearest Neighbor. After doing this exercise, you will certainly appreciate the complexity of target functions that can be learned with nearest neighbor!

2. **Decision trees.** (8pts) Throughout the course, we usually rely on implementations of machine learning algorithms in Python's scikit-learn library. This homework problem is very different: you are asked to *implement the ID3 algorithm for building decision trees yourself*. Refer to p. 56 in Mitchell for pseudocode of the ID3 algorithm that you are expected to implement. You are not allowed to use `sklearn` or any other library with a built-in implementation of ID3.

Your program should read in a training dataset, a test dataset (both as csv files), and the name of the target variable (= classification attribute), and output to the screen:

- your decision tree in a readable format (see below)
- its accuracy over the test set

Two pairs of sample datasets are available on Canvas, in the folder Files/homeworks/hw2, namely playtennis_train.csv and playtennis_test.csv, and republican_train.csv and republican_test.csv. The first line of the csv files contains the names of the fields. The target variable is not necessarily in the last column.

Name your Python program hw2.py. It should take command-line parameters for a file with training data, a file with test data, and the name of the target variable. In particular, it should run correctly when executing the following commands at the command-line:

```
python hw2.py playtennis_train.csv playtennis_test.csv playtennis
```

and

```
python hw2.py republican_train.csv republican_test.csv republican
```

The file hw2.py in Files/homeworks/hw2 contains starter code that visualizes a tree and computes accuracy. It will produce some output when run for the playtennis data (using the exact same command and arguments as written above). The output is nonsense, in the sense that the tree is hard coded and not constructed based on the data. You need to remove the `tree = funTree()` statement from the body of the `id3(examples, target, attributes)` function, and write a correct body for this function yourself. This is the only part of the starter code that you are expected to touch. You can of course introduce your own additional functions as you deem appropriate. Your final program hw2.py should also work for the republican data and for other, similarly structured, datasets.

Important notes:

- Write your code in Python 3. Python 3 is the first version of Python in the history of the language to break backward compatibility. This means that code written for earlier versions of Python probably won't run on Python 3. I won't be able to run and grade your program if it is written in Python 2.x. Any version of Python 3.x should be fine.
- I will not test your code on datasets with continuous-valued attributes. Your implementation of ID3 can assume that all attributes are discrete-valued. You are expected to use information gain to guide the search for the best split attribute.
- If you are new to Python, then this homework will require you to invest some time in learning the language. As Python is a prominent language among data scientists, acquiring basic skills in Python is an integral part of our machine learning course. A useful reference book on Python is: "The Quick Python Book", 2nd edition, Naomi R. Ceder, Manning Publications, 2013. From the book: "This book is intended for people who already have experience in one of more programming languages and want to learn the basics of Python 3 as quickly and directly as possible."

Submit:

(a) a printout of your code in class, as part of the hard copy of this homework

(b) an electronic version of your file hw2.py on Canvas