

STAT 598Z: Homework 1

Due: 31st January 2012

1. This homework will contribute 10 points towards your final score.
2. Attempt as many problems as possible.
3. Only neatly handwritten solutions will be accepted. Alternatively you may use L^AT_EX to typeset your solutions.
4. Hand in your HW (including print outs of your source code) at the beginning of the class on 31st January 2012. Additionally source code (if any) should be emailed to **stat598z@gmail.com** **before** the assignments are submitted in the class. No late submissions will be accepted!
5. Program files should be named after the problem (e.g. solution to problem 1 should be problem1.py etc).

Problem 1 (2 pt) In this problem we will do some simple text manipulation using UNIX commands. Below when I use the phrase “a UNIX command”, it either means a single command or a series of commands chained via pipe.

- The LIBSVM format is widely used to store datasets. Each line of a LIBSVM file represents a sparse vector (data) and its corresponding label in the following format:

```
label idx:val idx:val idx:val
```

Use emacs or any other editor of your choice to create a file `dummy.txt` with the following content:

```
+1 1:2 2:4 4:3.2
-1 2:2.1 4:8.2 6:5.9
+1 1:-6 2:-4 4:2.2
-1 3:9.6 5:6.3 8:5.8
-1 1:2.98 3:5.6 7:8.1
+1 5:0.3 7:4.1
```

The way to interpret the contents of this file is as follows: The first line represents the vector $(2, 4, 0, 3.2, 0, 0, 0)$ and its label is $+1$. The second line represents the vector $(0, 2.1, 0, 8.2, 0, 5.9, 0)$ and its label is -1 and so on.

- Write a UNIX command to count the number of unique labels in `dummy.txt`.

- Write a UNIX command to display the smallest `idx` value in `dummy.txt`.
- Download the `a1a` dataset from <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/a1a>. Verify that the UNIX commands you wrote above also work for `a1a`.

Hint: Read the `man` or `info` pages for `cut`, `sort`, `uniq`, `head`, and `wc` commands on your UNIX machine.

Problem 2 (4 pt) In this problem, you are asked to Write a Python program which converts the distance in terms of miles and kilometers.

- Write a program which takes as input the distance in miles and output the distance in kilometers. Note that the input distance has to be non-negative, otherwise an error message should be output.
- Write a program which converts every integer in the range 0 to 100 miles to kilometers.
- Write a program which can convert from miles to kilometers or vice versa. Your program should first take as input a string. If the string is `m2k` then it should convert the input from miles to kilometer and if the string is `k2m` then it should convert the input from kilometers to miles. It should only accept distances which is larger or equal to 0. It should continue receiving user input until the user signals the intention to stop by inputting a `q`. A typical session using your program should look like this:

```
Please input m2k or k2m: m2k
Converting from miles to kilometers
Please enter distance in miles: 70
70 miles is 113 kilometers
Please enter distance in miles: -70
Error: Input distance has to be non-negative
Please enter distance in miles: q
Bye Bye
```

Problem 3 (4 pt) We will write a small game in Python. Your program must generate a random letter between 'a' and 'z'. Then it should ask the user to input a letter and provide feedback about user's guess. To make the game a bit more tricky, note that unless the guess is correct, the feedback of the program is true for 67% of time, while it is false for 33% of time. The game is best described by a user session (suppose the generated random letter is 'i'):

```
I am ready to play. I have a letter between 'a' and 'z' in mind.
Enter your guess: e
Sorry your guess is too low
Enter your guess: m
Sorry your guess is too high
```

```
Enter your guess: g
Sorry your guess is too high
Enter your guess: f
Sorry your guess is too low
Enter your guess: j
Sorry your guess is too high
Enter your guess: i
Bingo!
```