

```
import sys

#constants
DEBUG = True
INPUT_FILE_NAME_TRAIN = "ALL_AML_grow.train.csv"
OUTPUT_FILE_NAME_TRAIN = "ALL_AML_train.csv.normalized"
INPUT_FILE_NAME_TEST = "ALL_AML_grow.test.csv"
OUTPUT_FILE_NAME_TEST = "ALL_AML_test.csv.normalized"

def debug(logMsg):
    if DEBUG:
        print(logMsg)

#just for a little bit of house keeping/debugging, we will keep a running list of every line
that's been modified
modified_lines_train = []
modified_lines_test = []

def proc_file(input_file_name, output_file_name):
    #https://stackoverflow.com/questions/3277503/how-do-i-read-a-file-line-by-line-into-a-list
    with open(input_file_name) as f:
        modified_lines = []

        #open the file which will be the train output; we don't want to append
        out_file = open(output_file_name, "w")

        #create a list of lines, stripped of the newline
        content = [line.rstrip('\n') for line in f]

        firstValue = True
        firstRecord = True

        for line in content:
            if len(line) <= 2:
                continue

            modified_line = []

            int_strings = line.split(',')

            #don't normalize the ID numbers
            if firstRecord:
                modified_lines.append(line + "\n")
                firstRecord = False
                continue

            #print(int_strings)

            for s in int_strings:
                if (firstValue):
                    firstValue = False
```

```
        modified_line.append(int_strings[0]);

        #int_strings.pop(0);
        continue

    if(s.isspace() or s == ""):
        continue

    if(int(s) < 20):
        modified_line.append(str(20))
    elif (int(s) > 1600):
        modified_line.append(str(1600))
    else:
        modified_line.append(s)

    #join the normalized values together again with commas, append the stripped newline
    modified_lines.append(",".join(modified_line) + "\n")
    firstValue = True

out_file.writelines(modified_lines)

return modified_lines

modified_lines_train = proc_file(INPUT_FILE_NAME_TRAIN, OUTPUT_FILE_NAME_TRAIN)
modified_lines_test = proc_file(INPUT_FILE_NAME_TEST, OUTPUT_FILE_NAME_TEST)

#print(modified_lines_train)
```