

# Project 2

Submit Assignment

---

**Due** Mar 31 by 8am    **Points** 130    **Submitting** a file upload    **File Types** zip

---

## Data Classification

MUST work in pairs ... WILL TAKE A LOT OF TIME

### *Introduction*

In addition to the Python (**VERSION 3.xx**) program(s) you'll be producing for this project, you are expected to create and submit a complete and well-organized report in PDF format as explained later.

In the data preprocessing project, you generated two ARFF files with genes in columns and values normalized to  $\geq 20$  and  $\leq 16,000$ . [Click HERE to download a zip file](#) containing two slightly modified versions of your files to use for this project:

Training data in *ALL\_AML\_AllGenes.train.arff* :

**7072** attributes including ID and class attributes

**38** data samples: first **27** with class label "ALL" followed by **11** "AML" samples

Testing data in *ALL\_AML\_AllGenes.test.arff* :

**7072** attributes including ID and class attributes

**34** data samples: **21** "ALL" samples and **13** "AML" samples, intermixed

Create a copy of the training data file and name it *ALL\_AML\_SigGenes.train.arff* and use this copy to perform the data preprocessing steps (**Steps 1** and **2**) outlined next.

### *Step 1: Examining gene variation*

**a.** A gene fold ratio is the maximum gene expression value across all samples (in a specific gene column) divided by the minimum value (in the same gene column). This value is frequently used by biologists to assess gene variability. Compute the fold ratio for each gene in the file.

**b. (Include answer in report)** What is the largest fold ratio? How many genes have it?

**c. (Include answer in report)** What is the lowest fold ratio? How many genes have it?

**d. (Include table in report)** Produce a table like below which counts how many genes have fold ratio *value* in the following ranges

<b>Range</b>	<b>Count</b>
$value \leq 2$	??
$2 < value \leq 4$	??
$4 < value \leq 8$	??
$8 < value \leq 16$	??
$16 < value \leq 32$	??
$32 < value \leq 64$	??
$64 < value \leq 128$	??
$128 < value \leq 256$	??
$256 < value \leq 512$	??
$512 < value$	??

**e. (Include graph in report)** Graph the fold ratio distribution above in order to highlight any interesting patterns. **BE CREATIVE!**

## Step 2: Finding the most significant genes

Samples 1-27 in the training set have class label “ALL” while samples 28-38 have “AML”. Considering every gene column separately, let  $Avg_{class=ALL}$  and  $Avg_{class=AML}$  be the average gene expression values for samples having class=“ALL” and class=“AML”, respectively. Similarly, let  $Stdev_{class=ALL}$  and  $Stdev_{class=AML}$  be their standard deviations. **PS: When computing the standard deviation, use formula for sample standard deviation NOT population standard deviation (STDEV.S instead of STDEV.P in Excel).**

The *T-value* statistic is used to test the hypothesis that two attributes are correlated (in our case, how much a specific gene column correlates with the class label column). T-values range between  $-\infty$  and  $+\infty$ : a value near 0 is evidence for the null hypothesis suggesting no correlation while a value far from 0 (either positive or negative) is evidence for the alternative (i.e., that there is correlation).

T-values are computed as shown next, where  $N_{classX}$  is the number samples labeled with *classX*:

$$\frac{|Avg_{class1} - Avg_{class2}|}{\sqrt{\frac{(Stdev_{class1})^2}{N_{class1}} + \frac{(Stdev_{class2})^2}{N_{class2}}}}$$

A similar correlation measure from electrical engineering is known as Signal to Noise (S2N) ratio which is defined as:

$$\frac{|Avg_{class1} - Avg_{class2}|}{Stdev_{class1} + Stdev_{class2}}$$

**a.** For each gene column, compute the average and standard deviation for both classes ("ALL" and "AML")—i.e.,  $Avg_{class=ALL}$ ,  $Avg_{class=AML}$ ,  $Stdev_{class=ALL}$  and  $Stdev_{class=AML}$ . Then, compute the T-value and Signal to Noise ratio for each gene. **Ignore genes with standard deviation of 0 for both classes** (i.e.,  $Stdev_{class=ALL}=0$  AND  $Stdev_{class=AML}=0$ )


For parts (b), (c) and (e), please consult file [Top50Genes.docx](#)  to check your answers

**b. (Include answer in report)** List the top 50 genes with the highest S2N ratio along with their S2N values.

**c. (Include answer in report)** List the top 50 genes with the highest T-value along with their T-values.

**d. (Include answer in report)** How many genes are common/in the **intersection** of (b) and (c) (i.e. exist in BOTH lists)? Show the number and an **alphabetized** list of matching genes in your report. **(Answer: should be 31 genes in common).**

**e. (Include answer in report)** How many genes are in the **union** of all genes from steps (b) and (c) above (i.e. include a gene if it occurs in EITHER steps (b), (c) or both). Show the number and an **alphabetized** list of matching genes in your report. **(Answer: there should be 69 such genes).**

**f.** Update *ALL\_AML\_SigGene.train.arff* to contain the genes from (e) (**not (d) as originally stated**) only by eliminating all others. Use the correct list of genes noted in [Top50Genes.docx](#)  even if your answers do not match.

**g.** Create a new file for the test data called *ALL\_AML\_SigGene.test.arff* which includes only the genes from (e) (**not (d) as originally stated**).

**h.** At this point, you should have a total of four files (2 training and 2 testing). Eliminate the ID column from each of the 4 files (as it won't be useful for classification purposes) and save the files

original: *ALL\_AML\_AllGenes.test.arff* and *ALL\_AML\_AllGenes.train.arff*

preprocessed: *ALL\_AML\_SigGene.test.arff* and *ALL\_AML\_SigGene.train.arff*

### Step 3: Building the classifier

Write your own *k*NN algorithm implementation in **Python (VERSION 3.xx)**, to work on this gene data. Your program should run smoothly ON ANY DATASET formatted in ARFF, which contains only numeric data along with a 2-valued class label (such as Yes/No or ALL/AML). Your program should take a training ARFF file and a testing ARFF file as input.

You will conduct an empirical study by trying out different distance/similarity measures and values for *k* in an attempt to determine what works best for the dataset at hand. Here are the required steps (please read all BEFORE you start writing your program):

**a.** Compute 4 different distance/similarity measures between each test sample and every training sample. Use the following measures: *Euclidean distance*, *Chebyshev distance*, *City-block/Manhattan distance* and *cosine similarity* (note that latter is a **similarity** NOT a distance measure).

**b.** Using each computed measure separately, find the *k* nearest neighbors for different values of *k* (try 3, 5, 7, 9 and 11)

**c.** For each distance/similarity measure and *k* value combination, predict the class for each test sample using weighted majority voting where each of the *k* neighbors gets a vote weighted at  $\frac{1}{dist^2+1}$  when using distance measures or  $sim^2$  when using similarity measures. In total, each test sample will get 20 predictions: 4 different measures \* 5 different values for *k*

**d. (Include in report)** For each distance/similarity measure and *k* value combination, create a confusion matrix and compute precision, recall and F-1 measure per class label.

**e. (Include in report)** Complete three tables like the one below to show the precision, recall and F-1 measures, respectively, for each class label (i.e., you will need six tables in total: 3 for ALL and 3 AML).

	<i>Euclidean</i>	<i>Chebyshev</i>	<i>City block</i>	<i>Cosine Similarity</i>
<i>k</i> =3	?	?	?	?
<i>k</i> =5	?	?	?	?
<i>k</i> =7	?	?	?	?
<i>k</i> =9	?	?	?	?
<i>K</i> =11	?	?	?	?

PS: ? above are precision, recall and F1-measure or values for a specific class label

**f. (Include in report)** Plot your tables from (e) using nice graphs in order to help your reader interpret the results. BE CREATIVE!

**g. (Include in report)** Conduct the above experiment twice each time using one of the two datasets: *original* and *preprocessed* (you should produce twice the number of tables and graphs noted in this part; half of them using the *original* data files and the other half using the *preprocessed* data files)

**h. (Include in report)** Answer the following questions using F-1 measures only. Justify your answers.

For each dataset above, what distance/similarity measure and  $k$  value combination are best for predicting *ALL* class?

For each dataset above, what distance/similarity measure and  $k$  value combination are best for predicting *AML* class?

Overall, which dataset is better for our purposes: *original* or *preprocessed*?

#### **Step 4: Submission**

EACH MEMBER of the group should make their own submission by the noted due date and time (replica of each other). Your submission will be in the form of a single zip file containing the following:

- 1. Report (in PDF format):** In addition to the items requested before, your report MUST include a 1-page description detailing what EACH MEMBER did along with a percentage value describing the member's share of the work (in an ideal world, each member gets 50%). If you disagree with your percentage, I ask that you inform me ASAP.
- 2. Final versions of files:** *ALL\_AML\_AllGenes.test.arff*, *ALL\_AML\_AllGenes.train.arff*, *ALL\_AML\_SigGene.test.arff* and *ALL\_AML\_SigGene.train.arff*
- 3. Complete program including ALL NECESSARY FILES** for me to run your work. **Also include a text file with instructions for me to run your program.** Your program should work for ANY DATASET formatted in ARFF which contains numeric/real data (NOT NECESSARILY INTEGERS) along with a binary class label (not necessarily *ALL* vs. *AML*). During your in-class demo, you will be asked to run your program on the given data files as well as ones that you haven't seen.