

<b>Due date:</b>	November 11, 2018
<b>Late submission:</b>	20% per day
<b>Teams:</b>	Students registered in COMP 472 can do the project individually or in teams of 2. Students registered in COMP 6721 must do the project individually. Teams must submit only 1 copy of the project.
<b>Purpose:</b>	The purpose of this project is to make you experiment with machine learning.

For this mini-project, you will use a machine learning framework to experiment with different machine learning algorithms and different data sets. The focus of this mini-project lies more on the experimentations and analysis than on the implementation.

### ML Framework:


You can choose between 2 different frameworks:

- WEKA** is an open-source machine learning package in Java (see <http://www.cs.waikato.ac.nz/ml/weka/>). Weka is available on Windows in the labs. It includes an easy to use graphical user interface to run and experiment with various algorithms and datasets. A nice tutorial on Weka is available at the URL: <https://www.cs.auckland.ac.nz/courses/compsci367s1c/tutorials/IntroductionToWeka.pdf>
- scikit-learn** is an open-source machine learning library for Python (see <http://scikit-learn.org/stable/>). scikit-learn is available on Unix in the labs. It is a more recent framework that provides an interface to program with a variety of different algorithms and built-in datasets. There are plenty of online documentation and examples of code.

### Data Sets:

You must use the 2 datasets provided on Moodle (see the zip file **DataSet-Release1**).

Both datasets are about the classification of black & white images of size 32x32 that represent a character.

For example, the image  represents the character 'A'.

Dataset 1 contains images of the 26 uppercase letters [A – Z] and 25 lowercase letters [a – z] (note that there are no instances for the letter 't').

Dataset 2 contains images of 10 Greek letters.

Each character in the datasets is represented by an index, as indicated in the table below:

Dataset 1								Dataset 2	
index	char.	index	char.	index	char.	index	char.	index	char.
0	A	10	K	20	U	30	e	40	o
1	B	11	L	21	V	31	f	41	p
2	C	12	M	22	W	32	g	42	q
3	D	13	N	23	X	33	h	43	r
4	E	14	O	24	Y	34	i	44	s
5	F	15	P	25	Z	35	j	45	u
6	G	16	Q	26	a	36	k	46	v
7	H	17	R	27	b	37	l	47	w
8	I	18	S	28	c	38	m	48	x
9	J	19	T	29	d	39	n	49	y
								50	z

Each dataset is in `.csv` format, where each row is a data instance.  
Each instance is composed of 1024 binary features followed by its class (the index).

Each dataset contains 3 splits:

- training: to be used for training your models.  
There are 1960 training instances in dataset1 and 6400 in dataset2.
- validation: to be used for validating/experimenting with your models.  
There are 514 validation instances in dataset1 and 2000 in dataset2.
- test: to be used to report your final output. These files are not included in **DataSet-Release1**.  
They will only be available a few days before the due date. These files will have the same format as the above but without the last column (i.e. they will only contain the 1024 features without the class at the end).

### Your task:

Run at least 3 different ML algorithms on the 2 datasets above for a total of at least 6 experiments.

For the algorithms, you must use:

1. a Decision Tree (DT)
2. a Naive Bayes classifier (NB)
3. a third algorithm of your choice. You can use one that we have not seen in class, as long as you explain in the report how it works.

Write the necessary code to generate one output file for each of your 3 models.

For example, if a test set called **ds1Test.csv** contains 3 instances:

<pre>0,0,1,1, ... // 1024 binary features 0,1,1,1, ... // 1024 binary features 0,0,0,1, ... // 1024 binary features</pre>
---

Your code should generate 3 files named

1. **ds1Test-dt.csv** // the classifications according to the Decision Tree
2. **ds1Test-nb.csv** // the classifications according to Naive Bayes
3. **ds1Test-3.csv** // the classifications according to your 3<sup>rd</sup> model

The format of these files should contain the row number of the instance, followed by a comma, followed by the index of the predicted class of that instance, as in:

<pre>1,24 // if your model's predicted class for instance 1 is 24 (Y) 2,50 // if your model's predicted class for instance 2 is 50 (z) 3,4  // if your model's predicted class for instance 3 is 4 (E)</pre>
--

Report, analyse and compare the performance of each algorithm with the datasets given in the report (see below).

### Experimentations:

Note that this is a mini-project, not an assignment. The difference is that it is open-ended and in addition to the required work stated above, you are expected to be creative, perform additional experimentations and analyse the results of your experimentations. In particular, play with the hyper-parameters, or other things, in order to increase the performance of your classification.

Part of your grade will depend on your performance on the test sets [given only a few days before the deadline]. So go ahead and try to improve the baseline models.

### Deliverables:

The submission of the project will consist of 4 deliverables:

1. Any code that you wrote
2. Your output files
3. A report
4. A demo

### The Code:

Submit all files necessary to run your code in addition to a **README.txt** which will contain specific and complete instructions on how to run your experiments on the desktops in the computer labs. You do not need to submit the datasets.

If the instructions in your readme file do not work, are incomplete or a file is missing, you will not be given the benefit of the doubt.

### Output Files:

Generate one output file for each of your 3 models and each of the validation and test sets. So in total, you should generate 12 files, named:

- [1-3] **ds1Val-dt.csv**, **ds1Val-nb.csv** and **ds1Val-3.csv**
- [4-6] **ds1Test-dt.csv**, **ds1Test-nb.csv** and **ds1Test-3.csv**
- [7-9] **ds2Val-dt.csv**, **ds2Val-nb.csv** and **ds2Val-3.csv**
- [10-12] **ds2Test-dt.csv**, **ds2Test-nb.csv** and **ds2Test-3.csv**

### The Report:

Write a report (~8-10 pages) to describe your experimentations and an analysis of the results:

1. The basic experimental setup (~2 pages):
  - Describe the algorithms that you chose to experiment with. Do not re-explain the theory of ML models we have seen in class, just indicate the hyper-parameters you used; and why you chose them. If you selected an ML model we have not covered in class, explain in a paragraph or two how it works and why you chose it.
  - If you experimented with various parameters, explain what you did and why you did it.
  - Describe any additional code that you may have written.
2. Analysis of the results (~5 pages)
  - State the results of all your experiments. A table would be a good format here. For each method and for each data set, show the results.
  - Analyse your results. For example, does the same algorithm perform the same way for different data sets? Why? What if you change some hyper-parameters?
3. Conclusion and Future Work (1-2 pages)
  - Draw final conclusions from your experiments. Do these results surprise you or are they expected?
  - If you were to continue working on this project, what do you feel would be interesting to investigate? Are there questions that you would like to investigate more, if you had the time and the energy?
4. References (not included in the page count)

- If you used any external references (books, online material ...), then list them here in the proper format. Failure to give proper references constitutes plagiarism.

The report must be in PDF format and must be called:

- **472\_Report2\_StudentID1\_StudentID2.pdf** (for team work) or
- **472\_Report2\_StudentID.pdf** (for individual work in COMP 472) or
- **6721\_Report2\_StudentID.pdf** (in COMP 6721)

Students in COMP 6721 (especially thesis students) are strongly encouraged to write their report in Latex (see <https://www.overleaf.com>).

### The Demo:

All submissions will be demoed during the lab time on the lab machines. You will not be able to demo on your laptop. Regardless of the demo time, you will demo the program that was uploaded as the official submission on or before the due date. The schedule of the demos will be posted on Moodle. No special preparation is necessary for the demo (no slides or prepared speech). Your TA will ask you questions on your code, and you will have to answer him/her.

### Evaluation Scheme:

	COMP 472	COMP 6721
Implementation (functionality, design, programming style, ...)	15%	10%
Demo (clear answers to questions, knowledge of the program, ...)	15%	15%
Report (clarity and conciseness, depth of the analysis, presentation, grammar,...)	35%	25%
Experimentations (thoroughness, originality,...)	25%	35%
Results with the test sets	10%	15%
<b>Total</b>	<b>100%</b>	<b>100%</b>

### Submission:

The code and the report must be handed-in electronically by midnight on the due date.

1. Create one zip file, containing all necessary files to run your program, the README.txt file and your report. Remember that your report must be in PDF and must be named as indicated in the section "The Report" above.
2. Name your zip file:
  - **472\_Project2\_StudentID1\_StudentID2.zip** (for team work) or
  - **472\_Project2\_StudentID.zip** (for individual work in COMP 472) or
  - **6721\_Project2\_StudentID.zip** (in COMP 6721)
3. Upload your zip file at: <https://fis.encs.concordia.ca/eas/> as **project2**.
4. In addition, please bring a paper copy of your report to class on Monday November 12.

Have fun!