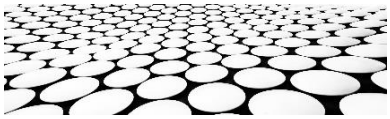
CSI4106 Introduction to Artificial Intelligence



PROJECT 1
Classification Empirical Study

GOALS

Supervised machine learning is a very empirical field, which means that we try many ideas to arrive at a solution, and we characterize such solution using the results obtained during our experiments. Within this empirical field, it is important that each machine learning study be reproducible, so that different people can arrive at the same results when using the same approach.

The overall goal of this project is to perform a classification empirical study and document it. More specifically, the goal is to get familiar with the experimental set-up required for a classification problem, as well as to further explore 3 classification algorithms/models seen in class.

At the end of this project, you will have:

- Reviewed your Python skills, as the project MUST be done in Python
- Explored and used a Python machine learning package, such as scikit-learn
- Explored Kaggle, an amazing resource of challenges and datasets
- Programmed 3 classification algorithms: Naïve Bayes, Logistic Regression, Multi-Layer perceptron
- Performed a classification empirical study using real data
- Documented, in a Jupyter Notebook, everything about your empirical study (view the Specific Requirements section), in a way to make your experiment reproducible

PLEASE NOTE: This is a demanding project. Do not wait until the last minute as you will not be able to achieve it. Also plan time for models to be learned which could take hours (during which you can do something else!).

SUBMISSION INFORMATION

- Deadline:
  - Report submission: **Tuesday, November 1st, midnight**
- Groups:
  - You are expected to form groups of 2 and do a single submission per group. You first need to register your group in Brightspace to later be able to do a group submission.
  - If you are 3 people working together, you have to form a group of 2 and a group of 1, work on different datasets and help each other out. You must mention this collaboration in your report.
  - If you prefer to work alone, that is fine, but the work is not reduced.
- Where to submit:
  - Your submission must be done in Brightspace in Assignment section (Project 1)
- Submission format:
  - No files accepted.
  - Your submission MUST be a **link** to a Colab Jupyter Notebook that the corrector will be able to go through (and run the code cells). If you prefer a different platform than Colab, that is fine, but the corrector MUST be able to access your notebook without having to install anything or copy any data.

PLEASE NOTE: If the corrector cannot access your notebook, or cannot run your code, the mark will be zero. It is your responsibility to test if your submission link works from a computer different than yours, as well as test that the cells in your Notebook are executable. You CANNOT submit a notebook file in Brightspace that the corrector would need to download, you must submit a link to a web-accessible notebook, ready to run.

TUTORIALS/TECHNOLOGIES

To achieve this project, you need to explore different environments. As this is a 4th year course, and you all have programming experience, I will let you do the exploration by yourself. You can use the Project 1 forum to share interesting resources for your colleagues and ask questions. Also, Baharin (your TA) has suggested the many links below.

What you need to know:

1. **Python**: The project MUST be done using the Python programming language. There are many tutorials that you can use to familiarize yourself with Python.

   - Python Tutorial for Beginners https://www.youtube.com/watch?v=t8pPdKYpowI
   - Python Full Course for Beginners https://www.youtube.com/watch?v=_uQrJ0TkZlc

2. **Scikit-learn package** https://scikit-learn.org/stable/ which contains several classification algorithms. You can use other packages if you prefer, but I recommend this one.

   - Scikit Learn Tutorial https://www.tutorialspoint.com/scikit_learn/index.htm
   - Machine Learning with Scikit-Learn https://www.youtube.com/watch?v=0Lt9w-BxKFQ
   - Machine Learning in Python Tutorial https://www.youtube.com/watch?v=pqNCD_5r0IU

3. **Jupyter Notebook**: Your project will have to submitted as a Jupyter Notebook. You can create/run such notebooks using Colab (see point 4).

   - Jupyter Notebook tutorial (Windows) https://www.youtube.com/watch?v=2WL-XTl2QYI
   - Jupyter Notebook tutorial (Mac) https://www.youtube.com/watch?v=HW29067qVWk

4. **Colab** https://colab.research.google.com for the creation of your final report (as a Jupyter Notebook) and to access machines that may be faster than yours for training classifiers. You do not have to train your models on Colab, you can work on your local machine, but... the training might be long.

   - Google Colab Tutorial https://www.tutorialspoint.com/google_colab/index.htm

REQUIREMENTS

1. Find a classification Dataset

You MUST choose a classification dataset from the Kaggle site. Make sure you are exploring datasets containing necessary data for a classification task (data annotated with classes). You can use this link to look for the classification datasets: https://www.kaggle.com/datasets?tags=13302-Classification Please note that you are not allowed to select the *Titanic dataset*.

If the topic of a dataset interests you, but the dataset seems too complex, it is always possible that you reduce the number of classes, and/or the number of training samples, and/or the number of attributes. You don't have to use the whole dataset, you can take a subset.

Also, although image classification is an interesting topic, it often requires models more complex than what we have learned so far, so I do not recommend using such dataset. If you really want to work with images, make sure that it's not a dataset of images of very high resolution, otherwise the learning time will be too long.

2. Perform a classification empirical study

1. Understand the classification task for your dataset
    a. Is it a binary/multi-class classification?
    b. What is the goal? Is this for a particular application?
2. Analyze your dataset
    a. Characterize the dataset in terms of number of training examples, number of features, missing data, etc.
3. Brainstorm about the attributes (Feature engineering)
    a. Think about the features that could be useful for this task, are they all present in the dataset? Anything missing? Any attribute provided that doesn't seem useful to you?
4. Encode the features
    a. As you will use models that need discrete or continuous attributes, think about data encoding and transformation.
5. Prepare your data for the experiment, using cross-validation
6. Train at least these 3 models using some default parameters. You should use ALL the models below:
    a. Naïve Bayes
    b. Logistic Regression
    c. Multi-Layer Perceptron
7. Test your 3 models using cross-validation (provided the split in step 5)
8. Perform an evaluation with precision/recall measures

9. For each type of model, modify some parameters, and perform a train/test/evaluate again. Do this for **two times**.
10. Analyze the obtained results
    a. Compare quantitatively (with the precision/recall measures) your 9 results. The 9 results come from 3 models, each with default parameters from step 6 + 2 variations from step 9.
    b. Show some examples of results that are good and not good (false positives and false negatives), try to understand why and discuss.

PLEASE NOTE: What is requested above is the minimum required for the project, you can do other tests by modifying the attributes, testing other models, etc.

3. Document your empirical study in a Jupyter Notebook

The purpose of the report is to illustrate the whole process followed during this project. Your Jupyter Notebook should include:

- Group number, names and student numbers of group members, report title
- If you collaborated between 3 people (Group of 1 + Group of 2) provide the link to the other Notebook
- A section for each aspect of the empirical study (10 steps mentioned above). If a section requires a Python code, add the Python code to a cell. If it requires an explanation or results, add them to a cell as well. Therefore, for each section, there will be either a python code (if it is a programming section), or an explanation/result cell.

PLEASE NOTE: As the learning is long to do, the corrector will not redo the learning, so please MAKE SURE that you KEEP THE CELL'S OUTPUTS for the submission so the corrector can witness what you did. I also suggest that some examples used in 10.b. for False Negatives and False Positives be included as examples the corrector could test and see the results (include code for doing so).

☆☆☆ EVALUATION

- Overall effort in the report (15%)
  - Writing in a clear and descriptive style that will allow the corrector to easily read/understand what was done, how and why
  - Good cell separation (text, code, results, etc)
  - Tests on various examples easy to perform by the corrector
  - Comparison between the approaches easy to understand (visualization using tables and/or graphs)
  - Report detailed enough for reproducibility
- Dataset choice justification + description (10%)
  - Justification of dataset choice
  - Description of the dataset & attributes (Steps 1 to 3 from empirical study)
- Experiment containing all steps that can be clearly followed (60% split as shown below)
  - Features correctly used (continuous/discrete) (10%)
  - Algorithms/models correctly programmed (25%)
  - Cross-validation correctly done (5%)
  - Evaluation correctly done (10%)
  - Variations on algorithms correctly done and explained (10%)
- Result analysis (15%)
  - Presentation of results (quantitative)
  - Selection of examples to illustrate false positives/negatives
- References (should be present, -10% if not)
  - For any part of your code taken from a web site (even a tutorial site or stackoverflow), you must provide the reference to it.
  - Any theory/algorithms found in books, slides, tutorials that you used should be referenced.

QUESTIONS

- You can ask your questions within the Project 1 topic of the discussion forum on Brightspace.
- You can also send an email to Baharin (balia034@uottawa.ca), but using the forum is a much preferred way as fellow students will benefit from your questions and Baharin's answers.