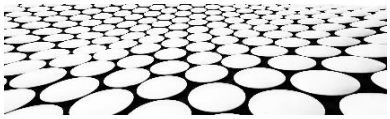
CSI4106 Introduction to Artificial Intelligence



ASSIGNMENT 2
Classification Empirical Study

Naïve Bayes vs Logistic Regression

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 assignment 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 2 classification algorithms/models seen in class.

At the end of this assignment, you will have:

- Reviewed your Python skills, as the assignment MUST be done in Python
- Explored and used a Python machine learning package, such as scikit-learn
- Explored Kaggle and UC Irvine Repository, as resources for datasets
- Programmed 2 classification algorithms: Naïve Bayes and Logistic Regression
- Realized 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 understandable and reproducible

PLEASE NOTE: This is a demanding assignment. Do not wait until the last minute as you will not be able to achieve it.

SUBMISSION INFORMATION

- Deadline:
  - Submission of link to your notebook: **Sunday, October 22nd, midnight**
  - Your notebook MUST NOT be modified following your submission
- 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.
  - As I want to allow groups to change at each assignment (if you want), I need to create a new set of groups for each assignment. You therefore need to register again.
  - If you prefer to work alone, that is fine, but the requirements are not changed.
- Where to submit:
  - Your submission must be done in Brightspace in Assignment section (Assignment 2)
- 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. Also make sure to provide the modification access which will allow correctors to view/modify your Notebook.

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 assignment, you need to explore different environments. You can use the resources suggested in Assignment 1 for Python, Jupyter Notebook and Colab. One additional resource, for experimenting with Naïve Bayes and Logistic Regression, is the Scikit-learn package.

Here are some references:

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

In particular, scikit learn contains methods for Naïve Bayes and Logistic Regression. You will find some examples of use of the models below.

Naive Bayes Classifier :

https://www.datacamp.com/tutorial/naive-bayes-scikit-learn

https://scikit-learn.org/stable/modules/naive_bayes.html

Logistic Regression :

https://www.datacamp.com/tutorial/understanding-logistic-regression-python

https://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html#logistic-regression-3-class-classifier

REQUIREMENTS

1. Choose a Classification Dataset

You **MUST** choose a classification dataset among the ones suggested below. All those datasets are for multi-class classification.

Glass Identification dataset:
https://www.kaggle.com/datasets/danushkumarv/glass-identification-data-set
Number of samples: 214, Number of attributes: 9, Number of classes: 7 (type of glass like tableware, headlamps, vehicle)

Dermatology dataset:
https://www.kaggle.com/datasets/olcaybolat1/dermatology-dataset-classification
Number of samples: 366, Number of attributes: 34, Number of classes: 6 (disorders -- psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris)

Maternal Health Risk:
https://archive.ics.uci.edu/dataset/863/maternal+health+risk
Number of samples: 1013, Number of attributes: 6, Number of classes: 3 (risk level – high, medium, low)

Car dataset:
https://archive.ics.uci.edu/dataset/19/car+evaluation
Number of samples: 1728, Number of attributes: 6, Number of classes: 4 (unacceptable, acceptable, good, very good)

Wine quality dataset:
https://www.kaggle.com/datasets/yasserh/wine-quality-dataset
Number of samples: 4898, Number of attributes: 11, Number of classes: 11 (0 to 10)

16 personalities dataset:
https://www.kaggle.com/datasets/anshulmehtakaggl/60k-responses-of-16-personalities-test-mbt
Number of samples: 60K, Number of attributes: 60, Number of classes: 16 (personality type)

Credit Score dataset:
https://www.kaggle.com/datasets/clkmuhammed/creditscoreclassification
Number of samples: 100K, Number of attributes: 27, Number of classes: 3 (Good, Standard, Poor)

2. Perform a classification empirical study

1. Familiarize yourself with the classification task and the dataset
    a. What is the goal of the task? Is this for a particular application?
    b. Characterize the dataset in terms of number of training examples, number of features, missing data, etc.

2. 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 feature provided that doesn't seem useful to you? Do you have the domain expertise to answer these questions? If you don't, think of ways to explore the attributes to establish whether they appear to influence the classification.
    b. What are the ranges of each feature? Try to compare and visualize those ranges. We discussed in class that attribute normalisation is often promoted as empirically helping to improve performances. Do you think this would be useful for your study?

3. Encode the features
    a. As you will use models that need discrete or continuous attributes, think about data encoding and transformation.
    b. Logistic regression:
        i. This classifier expects continuous attributes. You can use one-hot encoding for discrete attributes.
    c. Naive Bayes:
        i. We saw in class that this classifier expects discrete attributes. But in scikit-learn, there are several types of Naive Bayes classifiers, such as CategoricalNB and GaussianNB. The CategoricalNB matches what we saw in class (and includes an additional smoothing factor). But there is also GaussianNB which assumes a Gaussian distribution on continuous attributes. Choose one of the two classifiers according to your data. You can also test both (optional).

4. Define 2 models using some default parameters
    a. Logistic Regression Model: You will see that in scikit-learn there is a long list of parameters for logistic regression. Don't worry too much... just try to understand some parameters. We will discuss this again later in the semester. Also, later, in step 6, you will need to vary some parameters.
    b. Naïve Bayes Model: For the Naive Bayes model, there are also some parameters you can explore in step 6.

5. Train/test/evaluate your 2 models in cross-validation
    a. Use a 4-fold cross validation. Or (optional) you can explore by yourself the impact of using different values of k for k-fold.
    b. Perform an evaluation with precision/recall measures. Since you are looking at a multi-class problem, make sure that you compare micro and macro averages on

precision and recall. Discuss the differences (if any) obtained. For your particular dataset, are the classes balanced? That would impact the micro/macro results.

6. For each type of model (Naïve Bayes and Logistic Regression), modify some parameters, and perform a train/test/evaluate again. Do this for **two times**.
    a. **State clearly** what parameters you are changing using comments and a different cell per experiment. The parameters should be chosen in a way that they have an impact on the performance of the model. Parameters such as the number of iterations or "verbose" for logistic regression are not acceptable. For example, for Naïve bayes you can change the smoothing parameter. For Logistic Regression you can change the solver or tolerance. Explore changing other parameters (if any) than suggested.

7. Analyze the obtained results
    a. Compare quantitatively (with the precision/recall measures) your **6 results**. Your 6 results should use the same cross-validation technique (same k). The 6 results come from 2 models, each with default parameters from step 5 + 2 variations from step 6. Make sure to show your tests in cells. If you change a parameter, create a new cell and test. If you are making graphs for visualization, the values should not be "hardcoded".
    b. As was mentioned before, since you are looking at a multi-class problem, make sure that you compare with micro and macro averages on precision and recall. Discuss the differences (if any) obtained.

3. Document your empirical study in a Jupyter Notebook

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

- Group number, names and student numbers of group members, report title
- A section for each step of the empirical study (7 steps mentioned above). If a section requires Python code, add the Python code to a cell. If a section requires an explanation or results, add them to a cell as well. So, for each section, there will be either a python code (if it is a programming section), or an explanation/result cell, or a combination of cells for explanation + code. Don't put too much code in one cell. Practice making logical cell separations. For example, the definition of a function should be in one cell and its call in another.

☆☆☆

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 description (10%)
  - Justification of dataset choice
  - Description of the dataset, attributes and encoding (Steps 1 to 3 from empirical study). Clearly state what the attributes and target represent.
- 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 programmed and analyzed (10%)
  - Variations on algorithms correctly done and explained (10%)
- Result analysis (15%)
  - Presentation of results in a clear manner (comparative tables + **explanations**)
  - Simply mentioning "parameter X with the value 0.9 is better than with the value 0.8" is not enough. It is necessary to explain and/or provide hypotheses.
- 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 Assignment 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.