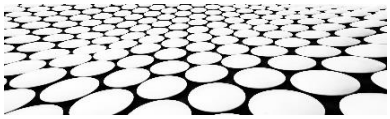
CSI4106 Introduction to Artificial Intelligence



ASSIGNMENT 4
Classification Empirical Study

Text Classification

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, we continue in the same spirit as Assignment 2 and further explore experimental set-up required for a classification problem, this time looking at deep learning approaches applied on textual data.

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 packages, such as scikit-learn
- Explored Kaggle as a resource for datasets
- Experimented with an MLP implementation (from scikit-learn or other)
- Experimented with simple NLP tasks with spaCy
- Performed a classification empirical study using real textual 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

SUBMISSION INFORMATION

- Deadline:
  - Submission of link to your notebook: **Sunday, December 3rd, 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 on Brightspace in Assignment section (Assignment 4)

- 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.

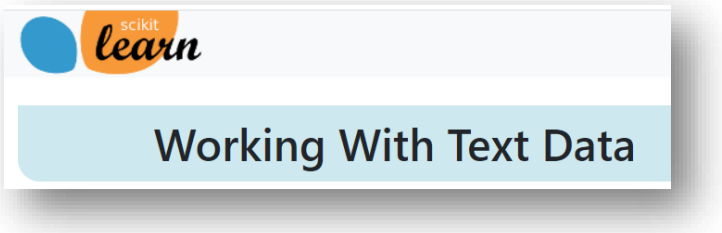
You can also use the resources provided in Assignment 2 for scikit-learn. You have already experimented with scikit-learn for Naïve Bayes and Logistic Regression.

Classification of textual data

In this assignment, you will explore using a MLP classifier for your classification task:

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Since you are working with textual data, you can look at the following tutorial:



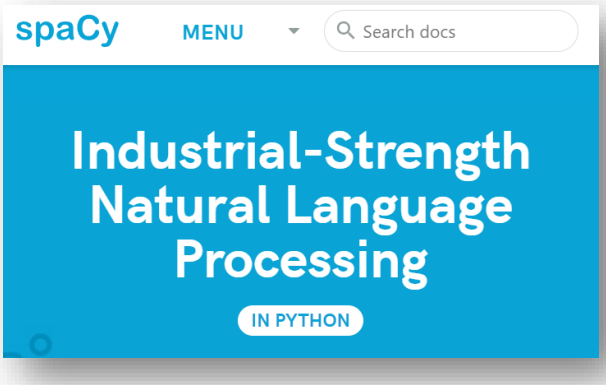
https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

Natural Language Processing

One additional resource, for performing simple NLP tasks such as tokenization, part-of-speech tagging and named entity recognition is spaCy (https://spacy.io/).

A Jupyter Notebook template is available in Brightspace to help you with this NLP part. I also encourage you to consult this section of spaCy which presents several examples of the use of "linguistic features".

https://spacy.io/usage/linguistic-features



REQUIREMENTS

1. Choose a Text Classification Dataset

You **MUST** choose a classification dataset among the ones suggested below. All those datasets are for multi-class classification. These datasets are larger than 25MB, so your TA has prepared reduced versions which you can use and upload on your Github.

- **CNN theme classification:**
  https://www.kaggle.com/datasets/pedroaribe/4000-cnn-articles-as-of-1062023
  There are 4000 articles to be classified into various themes (e.g. politics, sport, weather, …)

  Reduced version which has 1412 samples:
  https://github.com/baharin/CSI4106-Assignment4-Datasets/blob/main/reduced_file_cnnnews.csv

- **Airline Passenger Reviews:**
  https://www.kaggle.com/datasets/malharkhatu/airline-passenger-reviews
  64017 data samples. Classifies the passenger reviews into three classes (i.e., promoters, detractors, and passives).

  Reduced version which has 10761 samples:
  https://github.com/baharin/CSI4106-Assignment4-Datasets/blob/main/reduced_file_AirPassengerReviews.csv

- **Drug Reviews:**
  https://www.kaggle.com/datasets/jessicali9530/kuc-hackathon-winter-2018/data
  We will use the 'drugsComTest_raw' dataset. This is a large dataset with 53k samples with a lot of classes. We only consider the following classes in this assignment: Birth Control, Depression, Pain, Anxiety, Acne

  Reduced version which has 6398 samples:
  https://github.com/baharin/CSI4106-Assignment4-Datasets/blob/main/reduced_drugsComTest_raw_fiveclasses.csv

2. Perform some tasks from the NLP pipeline on textual data

To experiment with the NLP pipeline provided by spaCy, you will generate 2 additional datasets that will contain subsets of the textual information from your original dataset. We will call those datasets: *Derived-Dataset-1* and *Derived-Dataset-2*.

Derived-Dataset-1 must contain a subset of the POS (Part-of-Speech) tags. You can include only nouns, or only verbs, or only adjectives, or some combinations of those. The text should also be lemmatized. Depending on the dataset you chose, the role of different part of speech will be different, so I let you decide (and explain in your report) what you think is most important.

Derived-Dataset-2 must contain named entities found in the text + some POS that you think are important. You should explore what types of entities are extracted by spaCy and choose a subset that you think is appropriate for your task.

For example, a student group might have (if it's justified):

- Original dataset (text as is)
- *Derived-Dataset-1* with only the verbs and adjectives lemmatized
- *Derived-Dataset-2* with 3 types of named entities (organizations, money and dates) and with adjectives lemmatized

When you do your classification experiments, you will test with these 3 datasets and compare results.

3. Perform a classification empirical study

1. Encode the text as input features with associated values
    a. As you will be working with text data, you need to encode the text data as features. Take a look at the suggested tutorial in the Technology Section of this document to familiarize yourself with scikit-learn working with text data.
    b. Your text becomes a "bag-of-words", so the ordering is lost. Each word becomes an independent feature.
    c. You can remove stopwords in the original dataset to reduce the number of words.
    d. You can use the term-frequency (tf) or the tf-idf as the attribute value.

2. Define 2 models using some default parameters
    a. Logistic Regression Model: In scikit-learn there is a long list of parameters for logistic regression. For example, solver, max_iter, …
    b. Multilayer Perceptron Model: We will discuss in class some of the parameters for this model. You can have a look at MLPclassifier for a list of parameters.

3. Train/test/evaluate your 2 models (that have default parameters) on your 3 datasets. For each model,
    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.

4. Modify some parameters of the MLP model 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. Examples of modifications can be changing the hidden layer size, activation function or learning rate.

5. Analyze the obtained results
    a. Compare quantitatively (with the precision/recall measures) your **12 results**. Your 12 results should use the same cross-validation technique (same k). The 12 results come from 4 models (Logistic Regression + 3 variations of MLP) each applied on 3 datasets (Original + Derived1 + Derived2).
    b. 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".
    c. 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.

4. 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 describing how you derived your additional datasets. You should include your code along with its explanation and the factors that you considered for deriving the datasets. Explain your choice of POS and your choice of named entities.
- A section for each step of the empirical study (5 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 (20%)
  - 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 exploration / Creation of derived datasets (20%)
  - Description + Justification of the choices made to obtain the derived datasets
  - Code shows good understanding of the steps required to retain some specific POS and NE.
- Experiment containing all steps that can be clearly followed (50% split as shown below)
  - 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.