INF8245E(Fall 2021) : Machine Learning — Given on: Oct 25, 10:00 pm

Programming Assignment #3 — Due on : Nov 14, 10:00 pm

- This is an individual assignment. However, you are allowed to discuss the problems with other students in the class. But you should write your own code and report.
- If you have any discussion with others, you should acknowledge the discussion in your report by mentioning their name.
- You have to submit the `pdf` copy of the report on gradescope before the deadline. If you handwrite your solutions, you need to scan the pages, merge them to a single `pdf` file and submit.
- All the hyper-parameter details and other reporting details must be in the `pdf` report file only. You will have three types of files: 1) a single report (`pdf`) 2) dataset files (`.txt`) 3) a single code file (`.ipynb`). Submit all the text files and codes compressed to a single file `<your-matricule-ID>.zip` on **Moodle**.
  Note: gradescope doesn't accept .zip.
- Be precise with your explanations in the report. Unnecessary verbosity will be penalized.
- You are free to use libraries with general utilities, such as matplotlib, numpy and scipy for python. You can use pre-existing implementation of the algorithms available in scikit-learn package. Do not use NLTK or any other NLP libraries for pre-processing.
- If you have questions regarding the assignment, you can ask for clarifications in Piazza.

Medical Text Classification [50 marks]

In this assignment, we will build a classifier with medical-NLP corpus. This is a classification task with the input as a medical transcription (text) and the output as the corresponding medical transcript type. This is a clinical dataset which consists of a medical transcript from one of the 4 classes {Surgery - 1 , Medical Records - 2, Internal Medicine - 3 and Other - 4} as an input. The task is to classify the transcript (text) to the corresponding classes i.e the transcript_type. The dataset consists of 4000 transcripts in the training set and 500 in each of the validation and test sets. Link to the Dataset.

1. Most of the algorithms described in the class take the input as a vector. However, the reviews are natural language text of varying number of words. The first step would be to convert this varying-length movie review to a fixed-length vector representation. We will consider two different ways of vectorizing the natural language text: **binary bag-of-words representation** and **frequency bag-of-words representation** (as explained in the end of the assignment). Convert both datasets into these representations. Instructions for dataset submission are given in the end of the assignment (do not include the dataset in the report). [10 marks]

2. For this question, we will focus on the Medical-NLP dataset with binary bag-of-words (BBoW) representation. We will use the F1-score as the evaluation metric for the entire assignment. [19 marks]

   (a) As a baseline, report the performance of the random classifier (a classifier which classifies a review into a uniformly random class) and the majority-class classifier (a classifier which computes the majority class in the training set and classifies all test instances as that majority class). [2 marks]

   (b) Now train Naive Bayes, Decision Trees, Logistic regression and Linear SVM for this task. [Note: You should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for binary input features (also called Bernoulli naive Bayes).] [8 marks]

   (c) Report the list of hyper-parameters you considered for each classifier, their range, as well as the best values for these hyper-parameters, chosen based on the validation set performance[1].[3 marks]

   (d) Report the training, validation, and test F1-score for all the classifiers (with best hyper-parameter configuration). [3 marks]

   (e) Comment on the performance of different classifiers. Why did a particular classifier perform better than the rest? What was the role of the hyper-parameters in finding the best results. [3 marks]

3. Now we will repeat question 2 but with frequency bag-of-words (FBoW) representation. [21 marks]

   (a) Train Naive Bayes, Decision Tree, Logistic regression and Linear SVM for this task. [Note: Again, you should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for real valued input features (also called Gaussian naive Bayes).] [8 marks]

   (b) Report the list of hyper-parameters you considered for each classifier, their range, as well as the best values for these hyper-parameters, chosen based on the validation set performance. [3 marks]

   (c) Report the training, validation, and test F1-score for all the classifiers (with best hyper-parameter configuration). [3 marks]

   (d) Comment on the performance of different classifiers. Why did a particular classifier perform better than the rest? What was the role of the hyper-parameters in finding the best results. [3 mark]

   (e) Compare the performance with the binary bag-of-words based classifiers. Why is there a difference in the performance? Give a brief explanation comparing BBoW Naive Bayes and FBoW Naive Bayes and similarly for other models. [2 mark]

   (f) Which representation is better? Why? [2 mark]

[1] It is a good practice to discuss your hyper-parameter tuning in any report or paper. Whenever you report the performance of an algorithm, you should mention all the hyper-parameters, the range of values you considered for each of them, as well as the final values used to compute the test performance.

Instructions for binary bag-of-words representation

1. The first step is to construct the word vocabulary for a dataset. Given a dataset, we will only consider the training set and enumerate all unique words in the training set reviews[2]. To do this, we should do a bit of pre-processing to the raw text. For this assignment, we will do only 2 pre-processing steps: removal of punctuation, and lower-casing the words. For example, *(How old are you?)* would become *(how old are you)*.

2. Once you have the vocabulary, count the frequency of each word in the training set and sort the words in the vocabulary based on frequency (in descending order). Now pick the top 10,000 words in the vocabulary and ignore the rest of the words. These 10,000 words will form the feature set for our classification process.

3. For any example to be classified, generate a 10,000 dimensional feature vector as follows: For each of the top 10,000 words, there is one corresponding dimension in the feature vector that is 1 if the example contains the word, and 0 otherwise. Make sure to use the same pre-processing steps as before for validation and test reviews.

Instructions for frequency bag-of-words representation

1. Follow steps 1-2 of the instructions for the binary bag-of-words.

2. Now, for each of the 10,000 words, the corresponding feature is the frequency of occurrence of that word in the given review. You can calculate this by summing the occurrences of words in a review into a histogram, then dividing by the sum of occurrences of all 10,000 words so that the vector for each example sums to 1.

Instructions for dataset submission

For medical text classification dataset, you should submit medical_text-train.txt, medical_text-valid.txt, medical_text-test.txt, medical_text-vocab.txt.

1. medical_nlp-vocab.txt file should contain the 10,000 words in the vocabulary, their corresponding id (starting from 1), and their frequency. Each line represents a word, its numeric id, as well as its frequency, all tab separated. Example:

   the 1 20456

   where *the* is the word, 1 is the id of the word, and 20456 is the frequency of the word.

2. For train/valid/test file, each line is a data point. Each review should be represented as space separated ids for corresponding words in the review, with the class label at

[2]Think about why we should not include validation and test set in the vocabulary construction process.

the end, tab separated. Skip the ids for words which are not there in top 10,000 words. Just replace the word with the id (don't sort the ids in a row). Example:

100 8 3 1034 0

Here 0 is the class label and rest of the numbers represent a 4-word text.

Instructions for code submission

1. Submit a single zipped folder with your matricule id as the name of the folder. For example if your matricule ID is 12345678, then the submission should be 12345678.zip.
2. You can only use python and you must submit your solution as a jupyter notebook.
3. Make sure all the data files needed to run your code are within the folder and loaded with relative path. We should be able to run your code without making any modifications.

Instructions for report submission

1. Do not include your code in the report!