COMPSCI 361: Machine Learning
Assignment 3: Group Project (Worth 10% in Total)
Due date: 23:59 29 May 2023

General Instruction:

Recall that we have covered a series of supervised learning approaches for classification in our lectures, including **Naïve Bayes** (week 7), **kNN** (week 9), **SVM** (week 9-10), and **Neural Networks** (week 10-11). The goal of this assignment is to investigate supervised learning algorithms for article classification on BBC news datasets using NB, kNN, SVM and NNs. This assignment can be divided into four parts. Please write a python program to complete the following tasks using the Scikit-learn library:

- Task 1: Exploratory data analytics on text data
- Task 2: Perform classification models (NB, kNN, SVM, NNs) to build article classifiers using the given dataset.
- Task 3: Investigate the impact of multiple hyperparameters. Compare the classification quality across four classification models in terms of F1 measure. Learn to manage overfitting or underfitting situations.
- Task 4: Report your answers for each question and summarize your insights

Datasets:

Let's consider two classes of BBC news articles: tech news and entertainment news. After loading the data (.csv file) using pandas library, you will see that each news article is a comma-separated line with three columns: `news ID, processed news body, news class`. The processed news bodies are tokenized and lower-cased with removal of stop words and special characters. You can find the two data files, train.csv and test.csv, in `A3.zip` on Canvas. You can find examples of 2 approaches to pre-process the articles and extract features in the Jupyter Notebook `A3_processing_articles.ipynb`. We will discuss these methods briefly at the end of the lecture on Wednesday 10/05.

Submission:

Each group leader submits a single **report** ("`A3_Report_Group_#YourNumber#.pdf`" or in `.HTML`) and the **source code with detailed comments** ("`A3_Code_Group_YourNumber.py`" or in `.ipynb`) on Canvas by **23:59, Monday 29 May 2023**. Your report should be no more than five pages. You may submit to Canvas many times. Only the last version submitted before the deadline will be considered for marking. Each group member shall also fill the **peer review form** (link access with UoA credentials), including contribution weights of your own and the rest of the group members. The total contribution weights of your own and the rest of the group members should be added up to 1. Individual marks will be adjusted according to final weighting. If you do not fill the peer-review form, we will consider equal contribution.

Penalty Dates:

The assignment will not be accepted after the last penalty date unless there are special circumstances (e.g., sickness with medical certificate, family/personal emergencies). Penalties will be calculated as follows as a percentage of the mark for the assignment.

- By 23:59, Tuesday 30 May 2023 (10% penalty)
- By 23:59, Wednesday 31 May 2023 (30% penalty)

Advice about how to organize your work:

In this assignment, you will need to use knowledge ranging from preprocessing data, to evaluating classification models. The assignment is designed for you to use algorithms covered in the second part of the course (NB, kNN, SVM and NN).
Note that we will not cover SVM before week 9/10, and NN before week 10/11, and A3 is due beginning of week 12.
However, you can already complete the preprocessing Task 1, and Tasks 2, 3 and 4 for NB and kNN with the content we covered at the time the assignment is released.
We advise you start as soon as possible to work on these initial tasks. The pipeline will be very similar for SVM and NN, so it should be quick to run it for SVM and NN once we cover them in class. You will have enough time to complete everything if you start early!

Additional notes:

- It is perfectly fine to use the scikit-learn library in A3 tasks. You can also code your model if you prefer that way.

- An example of basic vectorization approaches you can use in Task 1 is included in the file *A3_processing_articles.ipynb*. These are common approaches to vectorize and extract features from text, but it is perfectly fine to have the dataset processed using other approaches of your choice. You may want to describe how you vectorize the articles in your report.

Task 1: Exploratory Data Analytics [1 pt]

(a) Load the dataset and construct a feature vector for each article in the entire dataset. You need to report the number of articles, and the number of extracted features. Show 5 example articles with their extracted features using a dataframe. **[0.5 pt]**

(b) Conduct term frequency analysis and report three plots: (i) top-50 term frequency distribution across the entire dataset, (ii) term frequency distribution for respective class of articles, and (iii) class distribution. **[0.5 pt]**

Task 2: Classification Models Learning [4 pts]

(a) **NB.** Train a Naive Bayes classifier using all articles features. Report the (i) top-20 most identifiable words that are most likely to occur in the articles over two classes using your NB classifier, and (ii) the top-20 words that maximize the following quantity:

$$\frac{\boldsymbol{P(X_w = 1|Y = y)}}{\boldsymbol{P(X_w = 1|Y \neq y)}}$$

Which list of words describe the two classes better? Briefly explain your reasoning. **[1 pt]**

(b) **kNN**. Train a kNN classifier on the training dataset. You need to report the surface plot of your kNN with your choice of hyperparameters *k* and distance metric. Explain the impact of *k* and the distance metric on the decision boundary. **[1 pt]**

(c) **SVM**. Train two SVM classification models (soft-margin linear SVM and hard-margin RBF kernel SVM) on the training dataset. You need to report two surface plots for: (i) the soft-margin linear SVM with your choice of misclassification penalty (C), and (ii) the hard-margin RBF kernel with your choice of kernel width (σ). Explain the impact of penalty C on the soft-margin decision boundaries, as well as the kernel hyperparameter on the hard-margin decision boundaries.

[1 pt]

(d) **NN**. Consider a neural network with the following hyperparameters: the initial weights uniformly drawn in range [0,0.1] with learning rate 0.01.

- Train a single hidden layer neural network using the hyperparameters on the training dataset, except for the number of hidden units (x) which should vary among 5, 20, and 40. Run the optimization for 100 epochs each time. Namely, the input layer consists of n features $x = [x_1, ..., x_n]^T$, the hidden layer has x nodes $z = [z_1, ..., z_x]^T$, and the output layer is a probability distribution $y = [y_1, y_2]^T$ over two classes.
- Plot the average training cross-entropy loss as shown below on the y-axis versus the number of hidden units on the x-axis. Explain the effect of numbers of hidden units.

$$CrossEntropyLoss = -\sum_{i=1}^{2} y_i \log(\widehat{y_i})$$

[1 pt]

Note for Task 2 (b) and (c)

For the surface plot, you can either (i) apply any dimension reduction (DR) techniques to project the data points into a 2-D plane or (ii) select two specific features that can effectively reflect the impact of the hyperparameters value. Explain your choices of features/DR techniques in your report.

Task 3: Classification Quality Evaluation [4 pts]

(a) We explore how the size of the training data set affects the test and train accuracy. For each value of m in [0.1, 0.3, 0.5, 0.7, 0.9], train your classifier on the first m portion of the training examples (that is, use the data given by XTrain[0:mN] and yTrain[0:mN]). Please report two plots: (i) training and (ii) testing accuracy for each such value of m with the x-axis referring to m and the y-axis referring to the classification accuracy in $F1$ measure as shown below. In total, there should be four curves for training accuracy and four curves for testing accuracy. Explain the general trend of the two plots in terms of training and testing accuracy if any.

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}$$

[2 pts]

(b) Let's use 5-fold cross-validation to assess model performance. Investigate the impact of key hyperparameters of your choices for each classifier using a testing dataset. E.g., for SVM, the classification accuracy may be significantly affected by the kernels and hyperparameter combination. List hyperparameters for each classifier and demonstrate how these hyperparameters impact on the testing accuracy. **[1 pt]**

(c) Report and compare your NB, kNN, SVM and NN classifiers with the best hyperparameter settings. Summarize what you have observed in the classification accuracy in $F1$ measure on the testing dataset. **[1 pt]**

Task 4: Report Writing [1 pt]

The mark associated with this task evaluates the structure and clarity of your answers and comments in previous tasks.

If you use Jupyter Notebook for this assignment, you may consider typing your answers and discussing directly in it and to export your notebook as an `.HTML` file and submit `HTML` and `ipynb` to Canvas.

Collaboration Policy:

- You should work on this assignment with the same group of students than the Tutorial project. Group members are responsible for dividing up the work equally and making sure that each member contributes. Please inform me early for mediations if you encounter any troubles in the group setting.
- The purpose of student collaboration is to facilitate learning. You are encouraged to seek help from each other in understanding the material needed to solve a particular homework problem. If you encounter difficulties working in group projects, feel free to consult with any of the instructors or tutors.
- You should try to solve problems you encounter within your group in priority, but do not hesitate to consult tutors or instructors if you are stuck on a problem for too long, or need any clarification.

Grading Rubric:

Task 1(a)	0.5 mark for the correct output.
Task 1(b)	0.5 mark for the correct output.
Task 2 (a)	1 mark for the correct implementation/use of NB and requested output.
Task 2 (b)	1 mark for the correct implementation/use of kNN and requested output.
Task 2 (c)	1 mark for the correct implementation/use of SVM and requested output.
Task 2 (d)	1 mark for the correct implementation/use of NN and requested output.
Task 3 (a)	1 mark for correct requested output format. 1 mark for the observations according to two requested plots with varying sizes of training data.
Task 3 (b)	1 mark for correct requested output format and the discussion of how chosen hyperparameters impact the testing accuracy.
Task 3 (c)	1 mark for discussion of comparative testing accuracy across four classifiers.
Task 4	1 mark for clarity of the report and clarity of the comments.