COL 774: Assignment 2. (Semester I, 2022-23)

Due Date: 11:50 pm, Tuesday Oct 4, 2022. Total Points: __

Notes:

- This assignment has two Main parts - Text Classification using Naïve Bayes (Part A) and MNIST Classification using SVM (Part B). There may possibly be an additional part comparing of Naïve Bayes and SVM on large scale text classification (Part C) - we will update on this soon. All parts will be due at the same time.
- You should submit all your code (including any pre-processing scripts written by you) and any graphs that you might plot.
- Do not submit the datasets. Do not submit any code that we have provided to you for processing.
- Include a **single write-up (pdf) file** which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- You should use Python for all your programming solutions.
- Your code should have appropriate documentation for readability.
- You will be graded based on what you have submitted as well as your ability to explain your code.
- Refer to the course website for assignment submission instructions.
- This assignment is supposed to be done individually. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. We will also include submissions from previous years since some of the questions may be repeated. Any cheating will result in a zero on the assignment, a penalty of -10 points and possibly much stricter penalties (including a **fail grade** and/or a **DISCO**).

1. **(26 points) Text Classification**
   In this problem, we will use the Naïve Bayes algorithm for text classification. The dataset for this problem is the Large Movie Review Dataset and has been obtained from this website. Given a movie review, the task is to predict the sentiment of the review as either positive or negative. Read the website for more details about the dataset. You have been provided with a subset of the Large Movie Review Dataset, with the training and test splits containing 25,000 reviews (samples) and 15,000 reviews respectively. Data is available at this link. A review comes from one of the two categories (class labels) —positive or negative.

   (a) **(10 points)** Implement the Naïve Bayes algorithm to classify each sample into one of the given categories.

      i. Report the accuracy over the training as well as the test set.

      ii. Read about word cloud. Construct a word cloud representing the most frequent words for each class.

   Notes:

   - Make sure to use the Laplace smoothing for Naïve Bayes (as discussed in class) to avoid any zero probabilities. Use $\alpha = 1$, where α is the parameter which controls the strength of the smoothing.
   - You should implement your algorithm using logarithms to avoid underflow issues.
   - You should implement Naïve Bayes from the first principles and not use any existing Python modules.

(b) **(2 points)**

i. What is the test set accuracy that you would obtain by randomly guessing one of the categories as the target class for each of the reviews (random prediction)?

ii. What accuracy would you obtain if you simply predicted each sample as positive?

iii. How much improvement does your algorithm give over the random/positive baseline?

(c) **(3 points)** Read about the confusion matrix.

i. Draw the confusion matrix for your results in parts (a) and (b) above (for the test data only).

ii. For each confusion matrix, which category has the highest value of the diagonal entry? What does that mean?

iii. In each of the cases above, the entries in the confusion matrix have a specific pattern. Can you observe this pattern and explain the reasons behind it?

(d) **(4 points)** The dataset provided to you is in the raw format i.e., it has all the words appearing in the original set of articles. This includes words such as 'of', 'the', 'and' etc. (called stopwords). Presumably, these words may not be relevant for classification. In fact, their presence can sometimes hurt the performance of the classifier by introducing noise in the data. Similarly, the raw data treats different forms of the same word separately, e.g., 'eating' and 'eat' would be treated as separate words. Merging such variations into a single word is called stemming. Read about stopword removal and stemming (for text classification) online.

i. Perform stemming and remove the stop-words in the training as well as the test data.

ii. Construct word clouds for both classes on the transformed data.

iii. Learn a new model on the transformed data. Report the test set accuracy.

iv. How does your accuracy change over test set? Comment on your observations.

(e) **(5 points)** Feature engineering is an essential component of Machine Learning. It refers to the process of manipulating existing features/constructing new features in order to help improve the overall accuracy on the prediction task. For example, instead of using each word as a feature, you may treat bi-grams (two consecutive words) as a feature.

i. You can read here about Bi-grams. Use word based bi-grams to construct new features. You should construct these features after doing pre-processing described in part d(i) above. Further, these should be added as additional features, on top of existing unigram (single word) based features. Learn your model again, and report test set accuracy.

ii. Come up with at least one additional set of features to further enhance your model. Learn the model after doing pre-processing as described in part d(i) above, and report the test set accuracy.

iii. For both your variations tried above, compare with the test set accuracy that you obtained in parts (a) and parts (d). Do the additional set of features constructed in each case help you improve the overall accuracy? Comment on your observations.

(f) **(2 points)** Read about Precision, Recall and F1-score.

i. For your best performing model obtained above (from parts (a), (d) and (e)), report the precision, recall and F1-score.

ii. Which metric, test accuracy or F1-score, do you think is more suited for this kind of dataset? Why?

2. **Remaining Parts of the assignment on SVMs, and possibly a comparison of the two algorithms, will be posted soon..**