ML Assignment-Week 1
Feb 28, 2024
Submission Date: March 01, 2024

==========Instructions==========

1. You are not allowed to use any external libraries other than scikit-learn, numpy, pandas, scipy, seaborn, and matplotlib. **Violating this rule will attract up to a 100% penalty.**
2. **Submission Format:** A zip file with the name **ML-Week1_<RollNo>.zip** containing 6 python files each specific to each task. Attach a ***pdf report*** in your submission containing description of hyperparameter tuning and final scores that you obtained.
3. Plagiarism will lead to **100% penalty** to all the involved parties. You are allowed to discuss among each other but no code artifact is to be shared among each other.
4. Scoring will be leaderboard based. Try to make the best-effort submission. Leaderboard will be made on a held out test-set and will be made public only after the deadline has passed.

==========

From this week onwards for the next two weeks, we are conducting assignments on Machine Learning and Deep Learning. The end goal is to build a classifier, which will label a social media post as either containing real and verified information or fake misinformation.

Dataset.
The dataset is from a shared task of Constraint@AAAI-2021. This task focuses on the detection of COVID19-related fake news in English. The sources of data are various social-media platforms such as Twitter, Facebook, Instagram, etc. Given a social media post, the objective of the shared task is to classify it into either fake or real news. It contains 10600 samples of which 5545 are labeled as real and 5055 are labeled as fake.

`If you take Crocin thrice a day you are safe.` **Fake**

`Wearing mask can protect you from the virus` **Real**

We now describe the specific tasks you are expected to do in this assignment**.**

Task-1: Prepare dataset.
Use scikit-learn train_test_split with shuffle option to split the dataset into training, validation, and test split. The fraction should be 80/10/10 (train/val/test). Dump the train/val/test split into three csvs and include it with your submission.

Task-2: Preprocessing Social Media Post.
Social media posts contain a lot of crucial information which is not in text-format like emojis, urls, hashtags. You need to carefully preprocess such social media text because filtering out emojis or hashtags might flip the stance of the post. You must go through the following tutorial to understand how you can preprocess social media text.

Task-3: Obtaining vector representations.
In general, inputs to machine learning models are vectors (and not say words). In practice, we use tf-idf representation to encode the input sentences to vectors which are then fed into ML models.

Task-4: Training ML classification models.

You are supposed to build the following basic ML binary classification models, that takes a social media post and classifies into 0:fake and 1:real.

1. K-nearest Neighbor
2. Logistic Regression
3. Support Vector Machine
4. K-Means Clustering
5. Neural Networks
6. FastText —This does not need tf-idf vectors, it directly takes raw text as input.

These ML models are very sensitive to the choice of hyperparameters that these ML models are initialized with. E.g., the K-Nearest Neighbor has two crucial hyperparameters: (i) ***K*** —the number of neighbors to consider, and (ii) The choice of metric to identify the ***K*** nearest neighbors.

You are expected to tune the hyperparameters involved in each of the Machine Learning Models. This might sound like a humongous task, but thankfully sci-kit learn provides you with a bunch of methods for hyperparameter optimization. This covers all models except for FastText for which you will have to write your own routine.

Task-5: Evaluating Machine Learning Models.

Now that your machine learning models are tuned and ready for evaluation, prepare a report that includes confusion matrix, classification accuracy, f1-score, precision, and recall on the test split that you obtained in Task-1. You can use the classification_report routine of scikit-learn for this task, but for sake of understanding we would recommend writing a routine of your own. Also include the final best set of hyperparameters obtained specific to each model in the report.