CDS Language Analytics Assignment #2: Binary Text Classification Benchmarking

Description

This repository aims to train 2 different binary classifiers (MLP & LR) using sci-kit learn. A vectorizer script is used to pre-vectorize the data once, such that the MLP and LR scripts don't have to do it again. The end result produces 2 metric reports of the different classifiers, as well as saving the models.

This code uses CodeCarbon to monitor the environmental effects of running this code. The effects of which can be found in the `out/emissions` folder

Setup

1. Make sure to have python and Git Bash installed!
2. Open a Git Bash terminal and use Git to download the repository:

```
git clone https://github.com/missingusername/cds-lang-git.git
```

3. Navigate to the project folder for this assignment:

```
cd cds-lang-git/assignment2
```

4. Before running the program, you first have to set up a virtual environment with the required dependencies. This can be done by simply running either `bash win_setup.sh` or `bash unix_setup.sh` depending on your system.
5. Before we can run the script, we first need the news data. The dataset can be found here, where you can download the `archive.zip` file. When downloaded, unzip the folder and place `fake_or_real_news.csv` in the `/in` folder of the `assignment2` directory.
6. To run the program, you can run the OS-appropriate run script. Here's an example of how to run the script with arguments:

```
bash win_run.sh
```

or

```
bash unix_run.sh
```

Takeaways from output

Logistic Regression	MLP

Class	Precision	Recall	F1-Score	Support
FAKE	0.84	0.81	0.82	628
REAL	0.82	0.85	0.83	639
Accuracy			0.83	1267
Macro Avg	0.83	0.83	0.83	1267
Weighted Avg	0.83	0.83	0.83	1267

Class	Precision	Recall	F1-Score	Support
FAKE	0.84	0.82	0.83	628
REAL	0.83	0.85	0.84	639
Accuracy			0.84	1267
Macro Avg	0.84	0.83	0.83	1267
Weighted Avg	0.84	0.84	0.84	1267

From these 2 tables we can see that both models actually end up performing remarkably well, and that it is hard to discern any real difference between their performances. However, we can notice that the `MLP` model does gain a small edge over the `LR` model in some areas, such as with its weighted avg. `F1-Score` being 0.1 higher.

Limitations & Improvements

While the reults seem to indicate the models being basically tied in terms of peformance, this could (potentially) be due to the train-test-split of the data providing equal results. To gain a more well-rounded picture of the models, one could use cross vaildation to see how the models would fare on average over multiple tries. This would of course also increase the power consumption and emissions of the scripts, since it would train the models e.g. 10 more times, just to see if the model training is reliable.