Machine Learning Challenge

Problem statement: Sentiment analysis in Twitter posts

Problem Overview:

In this machine learning challenge, participants will focus on the estimation of the sentiment conveyed by a Twitter post (tweet). The goal is to test participants' programming skills, creativity, and ability to implement various machine learning techniques for text analysis and classification.

Problem Description:

Background:

Tweets consist typically of text, emojis, hashtags, media, and links to external pages. Given their relatively short length, users are usually forced to be concise when conveying emotionally rich information; this means that, besides purely informative content and ironic text, tweets are usually easy to classify in pre-defined categories with respect to the sentiment they convey.

The task is to develop a solution for recognizing the emotional class of each tweet using machine learning algorithms.

Task:

The goal is to develop a solution for classifying tweets (text content) with respect to emotion classes using machine learning algorithms. Participants are required to implement the following tasks:

Data collection:

Find and download an annotated tweet database. You can start from <https://datasetsearch.research.google.com/search?query=Twitter%20Sentiment> or <https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis> or <https://www.kaggle.com/datasets/kazanova/sentiment140>

Make sure the data is annotated: each tweet should be marked with a ‘ground truth’ sentiment label. This can be numerical or text (emotion label).

Data cleanup:

Filter out links to external pages and media. You can choose whether to include emojis and hashtags; if you don’t want to use them as data, filter them out.

Classification using Supervised Learning:

Choose the input data for each tweet: you can assign individual emotion labels to each word or use bigrams or n-grams. You may need an emotion dictionary for either case, such as the NRC Word-Emotion Association Lexicon or something similar or use a sentiment analysis package to this end (e.g., NLTK, TextBlob, VADER).

Train and evaluate the performance of a supervised learning classifier. You can choose one of the following: k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Multilayer Perceptron (MLP) or try a recurrent NN (such as Elman or LSTM). Implement parameter tuning and choose the best classification strategy by performing a classification evaluation scheme based on the metrics of accuracy and confusion matrices.

Clustering using Unsupervised Learning:

Develop a clustering algorithm to identify the distinct clusters in the dataset. Choose a method that provides the best clustering results by performing a clustering evaluation scheme based on the metrics: Sum of Squared Errors (SSE), Cohesion, Separation, and Silhouette.

Deliverables:

Short Report: Describe in detail the architecture, algorithms used, and reasoning behind the chosen approaches for both supervised and unsupervised tasks.

Jupyter/colab Notebook or Python Scripts: Provide a well-documented Jupyter/colab notebook or Python scripts (\*.ipynb or \*.py) showcasing the implementation and results.

Submit both the report and your code to <https://eestec-athens-local-round-2024.devpost.com/> (account needed)

Additional Information:

Within this task, you are advised to use python. General-purpose Python packages that may be useful are: NumPy, Scikitlearn, Matplotlib, Pandas, TensorFlow, PyTorch, as well as sentiment analysis packages (e.g., NLTK, TextBlob, VADER).

Programming:

For the programming part, you can turn to StackOverflow if you bump into some kind of bug or problem, but again I strongly advise against copying code for these simple tasks. If you decide to use ChatGPT or something similar, make sure you test the code and add your comments.

As mentioned in the first tutorial, it is best practice to use Virtual Environments for your projects to avoid breaking the system. You can use either Virtual Environments of native python venv or install an Anaconda/Miniconda platform. Or simply use Google Colab.

Implement your approach using a Jupyter Notebook, with sufficient but not redundant comments.

Evaluation Criteria:

Projects will be evaluated based on the following criteria:

Data Pre-processing: Effectiveness of data augmentation and tweet clean-up techniques.

Supervised Learning: Performance of the chosen classifier in terms of accuracy and confusion matrices.

Unsupervised Learning: Clustering algorithm's effectiveness in identifying the distinct clusters.

In this challenge you will be tested on basic understanding of machine learning concepts, particularly supervised and unsupervised learning, as well as some Python and relevant libraries (e.g., NumPy, Scikit-learn, Matplotlib, Pandas, TensorFlow, PyTorch).