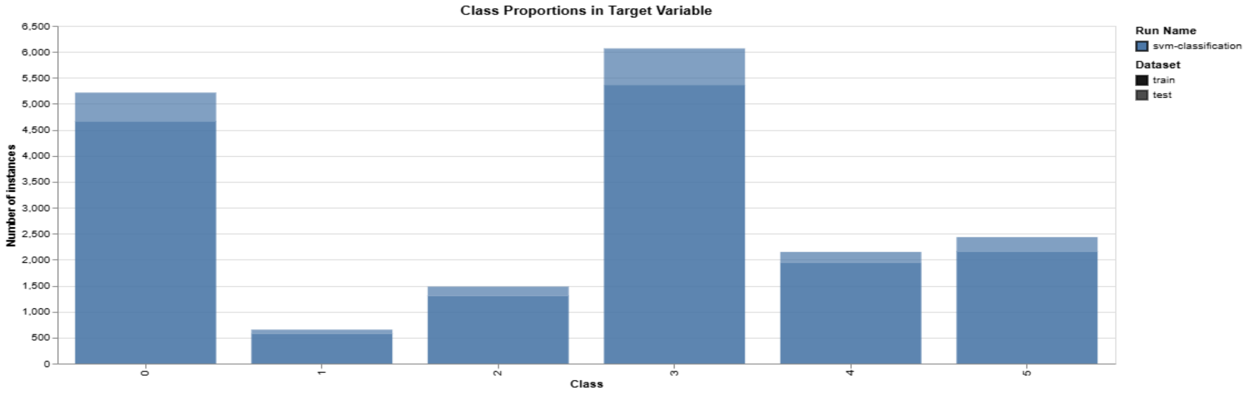
Practical Machine Learning

Preda Alexandru Florin

For this project, I have selected the "Emotion" dataset, which comprises 20,000 instances. The dataset is divided into three subsets: 16,000 rows are designated for training, 2,000 for validation, and the remaining 2,000 for testing. It includes six distinct emotion labels: 0 for sadness, 1 for joy, 2 for love, 3 for anger, 4 for fear, and 5 for surprise. This dataset will be used for both supervised and unsupervised learning tasks.

The class distribution within the dataset is relatively balanced, although there are some differences. The most populous class, joy (label 1), contains approximately 5,000 instances, while the least populous class, surprise (label 5), has around 500 instances. Despite these disparities, the performance scores across the classes were comparable, eliminating the need for techniques such as SMOTE or t-SNE to address class imbalance.



Data Processing:

For both tasks, the dataset underwent the same preprocessing steps:

1. **Lowercasing:** The entire sentence was converted to lowercase.
2. **Symbol Removal:** All non-alphabetical symbols were removed, leaving only the English alphabet.
3. **Tokenization (Optional)**: The sentence was tokenized into individual words using the *SpaCy* library with the *en_core_web_sm* model.
4. **Vectorization:** Three different approaches were tested to vectorize the sentences:
    a. **TF-IDF Vectorization:** The simplest method involved using TF-IDF to vectorize each sentence.
    b. **Word Embeddings (word2vec)**: The *word2vec-google-news-300* word2vec model was used to transform each word into 300-dimensional embeddings. The sentence was then represented as the average of these word embeddings along axis 0.

c. **Weighted Word Embeddings:** Using TF-IDF scores, a weighted average of the word vectors was computed to represent each sentence.

- **Supervised:**

For the supervised task, I have chosen two statistical models: Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). To train the models, I used *GridSearchCV* to find the best parameter combinations and *StratifiedKFold* for cross-validation with 4 folds. This approach helps prevent overfitting and yields more reliable results. The most representative score for this task is the weighted average F1 score.

Vectorization	KNN	SVM
TF-IDF	0.76	0.83
word2vec	0.48	0.70
Weighted word2vec	0.52	0.69

- **Unsupervised:**

For the unsupervised learning task, I selected two models: DBSCAN and K-Means. To optimize the performance of these models, I employed *GridSearchCV* to identify the best combination of hyperparameters. The effectiveness of the unsupervised models was then evaluated using the silhouette score, which measures how well clusters are defined.

Vectorization	DBSCAN	KMeans
TF-IDF	-0.0044	0.0081
word2vec	-0.1261	0.1221
Weighted word2vec	n/a	0.0548

Unfortunately, Weighted word2vec could not be clustered by DBSCAN algorithms with any combination of parameters.

- **What is being used in literature:**

In contemporary literature, the predominant models for text classification and generation are deeply rooted in deep learning algorithms. The majority of these models leverage Transformers, often combined with positional embeddings, to enhance performance. An exception to this trend is the MAMBA model, which represents a novel approach to implementing Transformers. If I were to select a deep learning model for sequence classification, my preference would likely be for BERT or another pre-trained model, which could then be fine-tuned for the specific task at hand.

- **Conclusion:**

In this project, I explored both supervised and unsupervised learning approaches to text classification, using the "Emotion" dataset as a case study. By experimenting with various preprocessing techniques and vectorization methods, I aimed to identify the most effective strategies for emotion detection. The results highlighted the strengths of traditional statistical models like SVM in supervised learning, particularly when combined with TF-IDF vectorization. Conversely, unsupervised models such as DBSCAN and K-Means presented challenges, especially in clustering tasks involving complex vector representations like weighted word embeddings. Despite these difficulties, the project underscores the importance of model selection and preprocessing in achieving robust performance. As the field continues to evolve, especially with the increasing adoption of Transformer-based models, further exploration and fine-tuning of these advanced techniques could offer even more promising results in text classification tasks.