Project Report: Author Classification Using Text Mining and Machine Learning Techniques

# 1. Introduction

The goal of this project is to identify the author of a given text based on their writing style. This is known as the author attribution problem, and it has applications in plagiarism detection, literary analysis, and forensic linguistics. The project utilizes a dataset consisting of articles from multiple authors and applies various text representation and machine learning techniques to perform classification.

# 2. Dataset and Preprocessing

The dataset is organized such that each author's texts are stored in separate folders. All `.txt` files under each folder are parsed and labeled accordingly. Preprocessing involved the following steps:

- Reading text files and assigning labels based on folder names (author names)

- Applying a cleaning function to remove excessive whitespace, special characters like `\*\*\*`, and normalize punctuation

- Label encoding of author names for classification

- Train-test split of 80%-20%

# 3. Feature Extraction Techniques

Four primary text representation techniques were used:

a. TF-IDF (Term Frequency-Inverse Document Frequency)

- Word-based unigram

- Word-based 2-gram and 3-gram

- Character-based 2-gram and 3-gram

These representations were implemented using `TfidfVectorizer` from `scikit-learn`.

b. BERT Embeddings

- `all-MiniLM-L6-v2` model from the `sentence-transformers` library

- Sentence embeddings were computed for each text

# 4. Classification Algorithms

Each feature set was evaluated using the following classifiers:

- Random Forest (RF)

- Support Vector Machine (SVM)

- XGBoost

- Naive Bayes (only with TF-IDF)

- Multi-Layer Perceptron (MLP)

- Decision Tree (DT)

Due to incompatibility, Naive Bayes was not applied to BERT embeddings.

# 5. Evaluation Methodology

Each model-feature combination was evaluated on the 20% test set using the following metrics:

- Accuracy

- Precision (weighted average)

- Recall (weighted average)

- F1-score (weighted average)

These metrics were obtained using `classification\_report` from `sklearn.metrics`.

# 6. Results Summary

Results were compiled into a CSV file (`results.csv`).

Highlights:

- Best TF-IDF performance came from character n-grams + SVM/XGBoost

- BERT embeddings worked well with SVM and MLP but performed worse than TF-IDF in this case, possibly due to dataset size or writing style similarity

# 7. Conclusion

This project demonstrated how different text representation techniques can significantly affect classification accuracy. While deep learning-based BERT models are powerful, traditional methods like TF-IDF paired with strong classifiers (SVM, XGBoost) often outperform in limited data contexts.

Prepared by: Osman Selim Yalçın

Submission Date: May 7, 2025