Table of Contents

[EDA for training.csv and test.csv 2](#_Toc210412108)

[Introduction 2](#_Toc210412109)

[EDA and Preparation 2](#_Toc210412110)

[Preprocessing 2](#_Toc210412111)

[Feature Extraction & Modelling 2](#_Toc210412112)

[Prediction & Results 2](#_Toc210412113)

[Conclusion 2](#_Toc210412114)

[Feature Selection 3](#_Toc210412115)

[Training the Model 3](#_Toc210412116)

[Interpret and Evaluate the Model 3](#_Toc210412117)

[Report 4](#_Toc210412118)

# EDA for training.csv and test.csv

## Introduction

This concise guide outlines the key steps for analysing and modelling two datasets: training.csv (ID, Text, Author) for learning, and test.csv (ID, Text) for prediction. The goal is to train a model that can predict the author of each test excerpt.

## EDA and Preparation

* Load both CSV files and inspect their columns and sample rows to confirm data structure.
* Check for missing values and remove incomplete rows.
* Examine text length and author frequency to spot outliers and class imbalance.
* Sample the excerpts per author for insight into writing styles.

## Preprocessing

* Convert text to lowercase and remove punctuation.
* Tokenize text and remove stop words.
* One-Hot encode target authors as numeric labels for modelling.

## Feature Extraction & Modelling

* Transform text with TF-IDF vectorization.
* Train a classifier on the training data.
* Use both RNN and FANs, mostly focusing to get the FAN accurate

## Prediction & Results

* Apply the same preprocessing and vectorization to test.csv.
* Predict authors for the test set, map numeric labels back to author names.
* Report predictions with ID and Author columns.

## Conclusion

This streamlined workflow provides the basics for text-based author prediction using machine learning. I’ll use tools like pandas and scikit-learn for efficient implementation and scalability.

# Feature Selection

Feature selection is crucial to improve model performance and reduce overfitting. To perform feature selection on both training and test data files, I’ll applying statistical methods such as chi-square or mutual information to identify the most relevant features from the TF-IDF matrix. I’ll select a subset of features that contribute most to distinguishing between authors and ensure the same feature selection process is applied to both datasets for consistency.

After I selected the top features, I’ll retrain the classifier using only these features on the training data and apply the same transformation to the test data before making predictions. This process helps streamline the model and can lead to better accuracy and generalisation.

# Training the Model

I trained a Recurrent Neural Network (RNN) model, which effectively captures text sequence patterns for author prediction. I designed the RNN architecture with optimal layers and hidden units, adjusting sequence length and applying dropout to prevent overfitting.

The RNN’s ability to model word order and stylistic nuances enhanced classification accuracy.

I also employed a Fuzzy Allocation Networks, which are particularly good at handling text classification tasks where class boundaries may be ambiguous or overlapping. During the training process, I tuned the network’s parameters, such as the fuzzification degree, allocation weights, and learning rate, using grid search or cross-validation to achieve optimal performance.

# Interpret and Evaluate the Model

To thoroughly assess the performance of the author prediction model, I will use a range of evaluation metrics. First, accuracy will give an overall indication of how many predictions match the true author labels. A confusion matrix will provide a detailed view of where misclassifications occur, guiding further refinement of the model. Additionally, I will use cross-validation scores to ensure the model’s results are consistent and not reliant on a particular train-test split.

# Report

Throughout this project, I systematically developed and refined a workflow for text-based author prediction using machine learning techniques. Initially, I focused on data preprocessing, ensuring that both the training and test datasets were cleaned and converted into a suitable format for analysis. This included standardising text, removing stop words, and applying TF-IDF vectorisation to transform the raw text into numerical features.

One of the key adjustments involved feature selection. By applying statistical methods I identified the most significant features from the TF-IDF matrix that contributed to distinguishing between different authors. This step was essential in reducing dimensionality and improving model generalisation. Importantly, I ensured that the same feature selection process was consistently applied to both the training and test datasets, maintaining the integrity of the evaluation.

For the model itself, I found that the RNN far outperformed the FAN model, both in accuracy and performance. During training, I performed parameter tuning, including adjusting the fuzzification degree, allocation weights, number of keywords and learning rate.

In evaluating the model, I relied on multiple metrics to obtain a comprehensive understanding of its effectiveness. Overall accuracy provided a broad indicator of success, while the confusion matrix offered insights into specific areas where misclassifications occurred.

Based on these findings and iterative adjustments, the final FAN model demonstrated worsened accuracy and generalisation, where the RNN model performed extremely well. The evaluation process highlighted the models strengths and weaknesses and pinpointed areas for potential future enhancement, such as incorporating additional features or exploring alternative classification algorithms.