EAST WEST UNIVERSITY

Stylometry-based-Author-Profiling-on-Bangla-Text-Using-Text-Clustering

A capstone project report to be submitted in partial fulfillment of the requirements for the degree

Of

Bachelor of Science in Computer Science & Engineering

by

Ishmam-ur-Rahman 2020-1-60-241 Computer Science & Engineering East West University 2020-1-60-241@std.ewubd.edu	MD Azman Ahmed 2020-1-60-217 Computer Science & Engineering East West University 2020-1-60-217@std.ewubd.edu
MD Suman Akanda 2020-1-60-266 Computer Science & Engineering East West University 2020-1-60-266@std.ewubd.edu	MD. Sanim Hossain 2020-1-60-267 Computer Science & Engineering East West University 2020-1-60-267@std.ewubd.edu

Under the supervision of

Dr. Mohammad Rezwanul Huq

Associate Professor

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Declaration

Project Title	Stylometry-based-Author-Profiling-on-Bangla-Text-Using-Text-Clustering
Authors	Ishmam ur Rahman, MD Azman Ahmed, Md Sanim Hossain, Md Suman Akanda
Student ID	2020-1-60-241, 2020-1-60-217, 2020-1-60-267, 2020-1-60-266
Supervisor	Dr. Mohammad Rezwanul Huq

We, the undersigned, hereby declare that the work presented in this capstone project is the outcome of the investigation performed by us under the supervision of Dr.Mohammad Rezwanul Huq, Associate Professor, Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh. This project has been presented to the department to fulfill the requirements for our Bachelor of Science in Computer Science and Engineering degree. We declare that this project has not been submitted elsewhere for any other degree or purpose. In addition, we acknowledge that our research received no personal or financial support.

Countersigned

Ishmam-ur-Rahman (ID: 2020-1-60-241)

MD Azman Ahmed (ID: 2020-1-60-217)

Md Sanim Hossain (ID: 2020-1-60-267)

Dr. Mohammad Rezwanul Huq

Md Suman Akanda (ID: 2020-1-60-266)

Dept. of Computer Science and Engineering
East West University
Dhaka, Bangladesh

Letter of Acceptance

This is to certify that the capstone project entitled Domain-Independent, User-centric Text Clustering Framework, submitted by Ishmam-ur-Rahman (ID: 2020-1-60-241), MD Azman Ahmed (ID: 2020-1-60-217), Sanim Hossain (ID: 2020-1-60-267) and Suman Akanda (ID: 2020-1-60-266) are undergraduate students of the Dept. of Computer Science and Engineering. Upon recommendation by the examination committee, we hereby accorded our approval as the presented work and submitted report fulfills the requirements for its acceptance in partial fulfillment for the degree of Bachelor of Science in Computer Science and Engineering.

Dr. Mohammad Rezwanul Huq
Associate Professor
Dept. of Computer Science and Engineering
East West University
Dhaka, Bangladesh

Dr. Maheen Islam

Chairperson, Associate Professor
Dept. of Computer Science and Engineering
East West University
Dhaka, Bangladesh

Date:

Acknowledgement

In the name of Allah, the Most Merciful, and the Most Compassionate, Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this capstone project. First and foremost, we would like to express our deep and sincere gratitude to our research supervisor, Dr. Mohammad Rezwanul Huq, for allowing us to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity, and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the East West University who taught us during our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Abstract

Bangla, the seventh most spoken language globally and the official language of Bangladesh, presents unique linguistic challenges with its 50 alphabets and compound words formed by combining multiple terms. While extensive research exists on authorship attribution in languages like English, Bangla remains underexplored, despite recent efforts in author profiling. One major obstacle is the scarcity of high-quality datasets, forcing researchers to create their own, often with suboptimal results. This study aims to address this gap by profiling authors of Bangla literature using stylometry-based techniques and unsupervised clustering. A comprehensive corpus of Bangla texts is processed through tokenization and text cleaning, followed by TF-IDF vectorization to capture unigram and bigram features. A two-step dimensionality reduction approach is applied, combining Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE), reducing the feature set to three dimensions for efficient clustering. K-Means and DBSCAN clustering algorithms are then used to identify distinct authorship patterns. The results are evaluated using external metrics, including Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Adjusted Mutual Information (AMI), and Fowlkes-Mallows Index (FMI), as well as internal metrics like homogeneity, completeness, and V-measure. Comparative analyses show clustering performance with and without dimensionality reduction, as well as alternative methods like t-SNE. The study's findings demonstrate the effectiveness of these unsupervised methods in Bangla author profiling, achieving 95.81% homogeneity, 94% ARI and NMI scores, and 93.45% completeness. These results provide a solid foundation for future multilingual stylometric analysis.

Keywords: Stylometry, Clustering, TF-IDF, t-SNE, ARI, NMI, FMI, Homogeneity, PCA, LSA, K-means, DBSCAN

Table of Content

List of Figures

List of Tables

1 Introduction

Bangla is the world's seventh most spoken language and the national language of Bangladesh. While there are many works on authorship attribution for other languages, particularly English, there are relatively few works for the Bengali language. People started working on it very recently! We had certain preferences in mind when selecting our capstone project. We were interested in a capstone project that would allow us to delve into the realm of artificial intelligence and machine learning. So, we decided to work on author profiling in Bangla literature. Author profiling involves a computational process in which a set of documents from predefined authors is trained to detect the correct authorship of unknown documents. There are many ways for profiling authors but the most favorable form for researchers is “stylometry”. Our methodology is primarily motivated by the notion that every author has a natural and unique way of writing and expressing their feelings. Similarly, stylometry operates under the assumption that each writer possesses a distinctive writing style, which can be discerned by analyzing specific linguistic features in their writing. The richness of this language presents both challenges and opportunities for author profiling. Stylometry proves to be a highly effective tool for profiling authors of literary works. By examining linguistic features such as word choice, sentence structure, and punctuation usage, stylometric analysis can unveil the unique writing style and preferences of an author. Machine learning algorithms, especially those trained on literary texts, can also serve as effective tools for author profiling in literature. Additionally, combining stylometry, machine learning, thematic analysis, and consideration of literary devices can further enhance author profiling for literature books. Ultimately, a blend of these methods may yield the most reliable and comprehensive results.

Our primary goal for this research is to apply stylometry-based techniques and text clustering methods to effectively attribute/profile authors of Bangla literature. Stylometry is a subfield of computational linguistics. It mainly focuses on analyzing the linguistic features of a text to distinguish each author's writing style. We will try to leverage advanced text clustering techniques to statistically analyze and identify and group authors based on their writing patterns. This research has the potential to enhance the accuracy and efficiency of the current authorship attribution processes and present valuable insight into the fields of digital forensics and literary analysis.

This step gives an overall outline of the methods we are planning to use throughout this research procedure. We will go through a research design, data collection, data cleaning, preprocessing, feature extraction, clustering algorithms, and evaluation metrics. We aim to develop a robust framework for authorship attribution that can be applied to diverse textual data.

1.1 Research Questions

The above-mentioned primary research question is then divided into several sub-research questions according to the Program Outcomes mentioned in the curriculum.

P01: How can machine learning, specifically NLP, be applied to analyze the writing styles of prominent Bangladeshi authors in Bengali literature for authorship determination?

P02: What specific stylometric features or markers can be identified and selected to effectively validate the authorship attribution system for Bengali literature?

P03: How can the author attribution system be designed to be flexible and not dependent on the length of the literary document, enabling its application to various modern writers?

P04: What size and diversity of a training corpus is needed to build an accurate author attribution system for Bengali literature?

P05: What machine learning classification models are most suitable for author attribution in Bengali literature, and how do they compare in terms of accuracy?

P06: What impact does the combination of multiple features have on the accuracy of the author attribution system?

P07: How does the proposed author identification system, incorporating visual information, compare to traditional feature-based strategies for authorship determination?

P08: What impact do varying data quantities and dataset splits have on the performance of the author attribution system in Bengali literature?

P09: How does the performance of the proposed system compare to existing works in the field of Bangla language authorship identification, and what factors contribute to its superiority?

P10: What cultural implications arise from the use of stylometric analysis in Bengali literature, and how does this impact the preservation of literary traditions?

P11: What are the social and ethical considerations associated with implementing stylometric analysis for authorship detection in a cultural and linguistic context like Bengali literature?

P12: How does the application of machine learning and NLP in author attribution impact the environment, considering aspects like data collection and processing?

1.2 Research Objective

Domain-Independent: One of the goals of this Capstone Project is to create a versatile Bangla text clustering framework that can be effectively used for author attribution across different domains. This framework is scalable and can handle large datasets.

User-Centric Approach: Another goal of the Capstone Project is to enable the text clustering framework to learn a writer's stylometry, providing users with accurate information about the author of a given piece of literature.

Contextual Analysis: In specific domains, such as politics, various contexts or dimensions may exist. One objective of our Capstone Project is to comprehend the context within an environment and deliver results accordingly. For instance, our innovative framework can understand the document's context and provide more relevant results to users. By applying clustering algorithms to the Bangla dataset, our framework can detect the author of a text based on the given context.

Contextual Similarity: While many robust algorithms can identify structural similarities, our focus is on developing a novel method to find contextual similarities between documents by building on and improving existing methodologies incrementally. This is another key objective of the Capstone Project.

Environment-Friendly: The Capstone Project aims to be eco-friendly. Our proposed framework maintains a large corpus of documents and delivers results electronically. Being a cloud-based service, the model operates without paper, thus reducing the need for manual pen-and-paper work and modestly decreasing carbon emissions.

1.3 Focus and Contribution

This project involves developing a Flask-based web application designed to process and analyze Bangla text. Users can input Bangla text through an HTML form, which is then preprocessed and prepared for clustering analysis using state-of-the-art machine learning techniques.

The focus of this project is on implementing advanced text processing methods and integrating them into a user-friendly web interface. The application leverages modern techniques in text vectorization, dimensionality reduction, and clustering to provide meaningful insights into Bangla literature.

Key contributions of this project include:

- **Efficient Data Processing:** Implementation of an end-to-end pipeline that takes raw Bangla text, processes it through vectorization techniques, and reduces dimensionality using PCA and t-SNE for better visualization and analysis.

- **Advanced Clustering Techniques:** Application of clustering algorithms such as K-Means and DBSCAN to categorize Bangla texts based on their contextual similarities.
- **Integration of Modern Technologies:** Utilization of advanced tools and frameworks, including Flask for web development and pre-trained models for text vectorization, to streamline the workflow and enhance performance.
- **User-Friendly Interface:** Development of an intuitive web interface that allows users to easily input text and receive clustering results, making complex text analysis accessible to non-technical users.
- **Contextual Understanding:** Improvement in the understanding of Bangla text through the application of sophisticated machine learning models, providing deeper insights into the structure and themes within Bangla literature.
- **Model Performance:** Evaluation and optimization of model performance to ensure accurate and meaningful clustering results, contributing to the field of text analysis in Bangla.

This project demonstrates the power of integrating machine learning with web technologies to create a robust platform for analyzing Bangla text, offering a valuable tool for researchers and enthusiasts in the domain of Bangla literature.

1.4 Organization of Book

The structure of our book is organized as follows: Chapter 1 introduces the project's principal goal. Chapter 2 summarizes existing work relevant to our research topic, describing the current state of the field, the study's main focus and limitations, and the research objectives. Chapter 3 presents the materials used and details the research methodologies, including design, implementation, and project management aspects. Chapter 4 analyzes and evaluates the model's results. Finally, Chapter 5 concludes the study and discusses potential future work. Figure 1 shows the flowchart of this book.

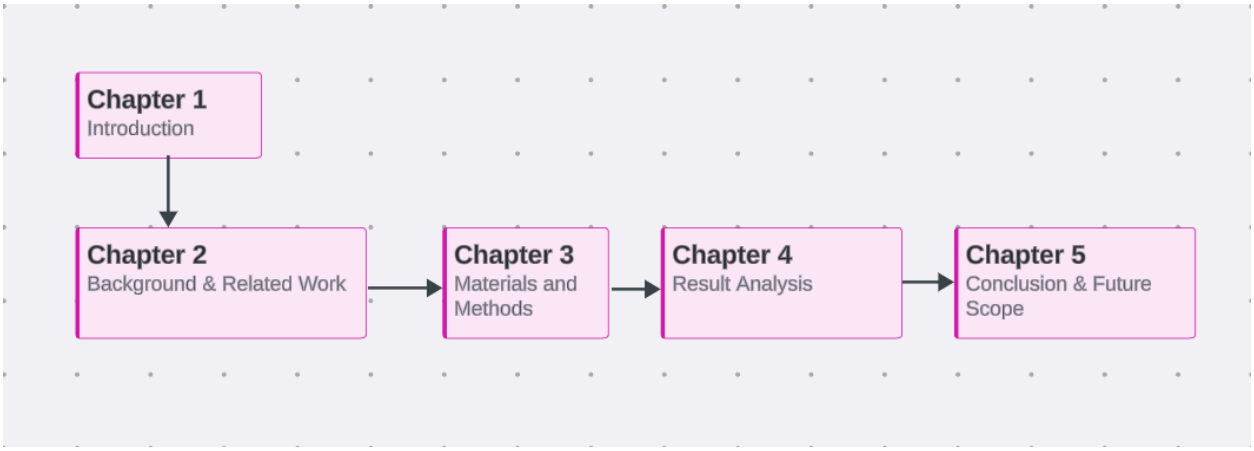


Figure 1: Organization of Book

2 Background and Related Work

2.1 Existing Methodologies

The main objective of our research is to use stylometry and text clustering techniques to accurately attribute and profile authors of Bangla literature. Stylometry, a branch of computational linguistics, focuses on examining linguistic features to identify individual writing styles. We aim to utilize advanced text clustering methods to statistically analyze, identify, and categorize authors according to their writing patterns. This approach could improve the precision and effectiveness of current authorship attribution methods and offer valuable insights into digital forensics and literary analysis.

This step gives an overall outline of the methods we are planning to use throughout this research procedure. We will go through a research design, data collection, data cleaning, preprocessing, feature extraction, clustering algorithms, and evaluation metrics. We aim to develop a robust framework for authorship attribution that can be applied to diverse textual data.

2.2 Literature Reviews

1. Hossain, A. S. et al., (2020) [1] proposed a method for determining the authorship of anonymous texts by analyzing the writing style of linguistic data. They aim to identify the true author of these texts with verification of authenticity. To achieve this, the authors have developed an author attribution system that analyzes the writing patterns of eight Bangladeshi bloggers. In total, the corpus includes 1764 articles, with each author contributing approximately 185 articles. A training set of 1399 articles and a testing set of 365 articles were used. The first step involves identifying the author, followed by counting the frequencies of pronouns and conjunctions for individual authors. The authors employed two classification models, namely Feedforward Neural Networks and Support Vector Machines, both of which are machine learning algorithms. The SVM classifier with a linear kernel achieved a 70% accuracy rate and performed better than the radial basis function and polynomial kernel variations. The model's average accuracy rates of 76.4% and 75% for the selected features. Authorship detection, based on statistical text analysis offers a promising solution to these challenges.

2. Hossain, M. T. et al. (2017) [2] proposed a method for Stylometric Analysis of Bengali Literature to determine authorship. In their research paper, they focused on the writings of six prominent Bangladeshi authors of the modern era. The most challenging feature of their work involved selecting appropriate style markers or features to validate the authorship attribution system. This system is especially flexible and doesn't depend on

the length of the document, making it applicable to other modern writers as well. In their classification model, the researchers employed a test corpus comprising 300 unknown documents, with 50 documents from each author, which were not utilized in the initial statistical analysis. Here, individual features exhibited an average success rate of 80%. When combined into a voting system with five features, this system achieved a 90.67% average accuracy rate in identifying the original author of an unknown document.

3. In this study, Islam, M. A. et al [3] explored authorship attribution in Bengali literature, specifically focusing on five Bangladeshi authors. A web crawler was developed using python code to scrape the articles of five renowned writers. They employed stylometric features, such as frequently used words, word length, sentence length, WH words, and numbers, to identify each author's unique writing style. An artificial neural network (ANN) model was developed, achieving over 85% accuracy for author identification. However, limitations include a limited dataset, potential feature selection improvements, ethical considerations, and the need for further model evaluation and comparison with other algorithms.

4. Dhar, A. et al. (2022) [4] proposed a convolutional neural network (CNN)-based author identification system from literary articles. For author identification, the system combines a five-layer convolutional neural network with visual information.1200 articles by 50 authors were the subject of experiments, with a maximum accuracy of 93.58%. Moreover, dataset splits were used in the trials to observe the system's performance on varying data quantities. The method achieved better results on publicly available datasets than both established works and traditional handmade feature-based strategies. They show that the maximum accuracy was obtained for the total dataset using the combination of both the statistical-based and text-based features.They compared different classification algorithms considered here as those for which the maximum accuracy had been obtained for the CNN. The results obtained are shown in **Figure.1**. They also compared the performance of the proposed system with other existing works performed in the Bangla language. Specifically Anisuzzaman et al. (2018) [5] proposed model shows the accuracy of the dataset is just 84.28%. The performance of the proposed model was better than all other existing algorithms when accuracy was concerned and the dataset used in this study was larger when compared to all the other research.

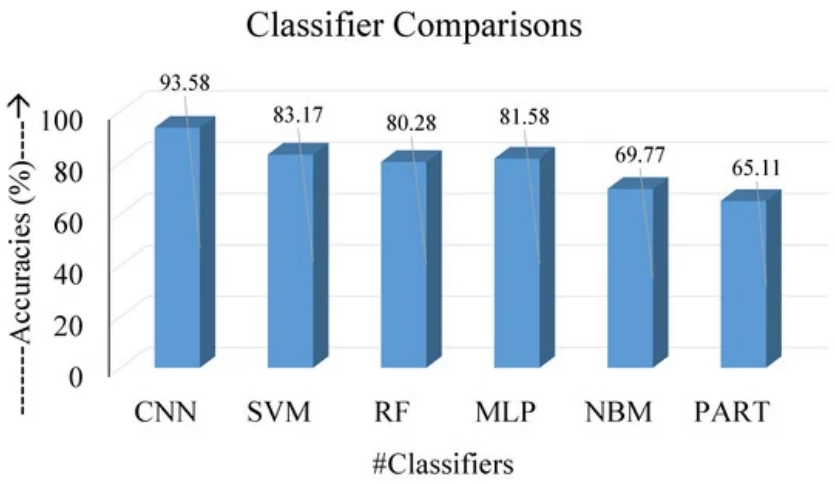


Figure 2. Performance of different popular classifiers.

5. While there are many works on authorship attribution for other languages, particularly English, there are relatively few works for the Bengali language. Anisuzzaman et al. (2018) [5] show the authorship attribution for three Bengali writers using both the Naïve Bayes method and a new method proposed by them which performs better than Naïve Bayes for authorship attribution. For their method, they combine both n-gram and Naïve Bayes methods and they find the bigram count for adjacent words and unigram count for the single words and combine the N-Gram algorithm with Naïve Bayes to formulate their proposed method. In this case, the accuracy provided by Naïve Bayes is 86%, whereas the accuracy provided by their technique is 95%. They created their dataset of 107380 terms and 21198 unique words for this experiment. To create their dataset for this experiment, they collected the Bengali literature of three different authors named: 'Humayun Ahmed', 'Rabindranath Tagore' and 'Shamsur Rahman'. After that, the dataset is transformed in some way. According to the experiment, their method works better than Naïve Bayes method.

6. Aisha Khatun et al. (2019) [6] proposed a CNN architecture based on character-level signals. According to them, character-level CNNs can be much more time and memory efficient. With enough training data, the accuracy is on par with the best performing world-level models. They used character embeddings to get meaningful numerical vector forms from characters. Where word embeddings require higher vector size (100-300), character embeddings only need a small vector size (e.g.: 16). They pretrained their model with a news dataset to learn the character embeddings. In their classification task, they used one-hot and pretrained embeddings. They compared their model with Word2Vector, and fastText (both CBOW and skip-gram versions). In comparison between these models and their proposed architecture, their architecture was performing reasonably well as long as the given data set was big enough.

7. Rajib et al. (2021) [7] proposed an authorship classification approach utilizing Convolutional Neural Networks (CNN). They worked with a dataset comprising 949,062 unlabelled texts and 18 authors with 25,749 labeled texts. Their algorithm employed Support Vector Machine (SVM) and Naive Bayes (NB) in addition to CNN. Impressively, CNN achieved the highest accuracy rates of 93.45% for BACC-18, 95.02% for BAAD16, and 98.67% for LD. The study contributed by creating new resources, evaluating 90 embedding models, optimizing 36 classification models, and achieving notable accuracies. However, it faced limitations such as restricted generalizability, reliance on embedding techniques, and a narrow focus on accuracy metrics.

8. Khan, T. F. et al. (2023) [8] developed a Convolutional Neural Network (CNN) architecture aimed at authorship verification, focusing on Urdu text. Their dataset comprised over 21,918 documents sourced from newspaper websites. Employing CNN with hyper-tuned parameters, they trained and evaluated models on the UAVC-22 dataset, achieving an impressive 93% accuracy rate in verifying Urdu text authorship. Their contribution included the creation of an extensive Urdu corpus and the introduction of nine embedding models. However, they noted limitations in computational complexity and challenges associated with CNN decision-making processes.

9. Pal, U. et al(2017) [9] proposed a machine learning approach for stylometric analysis of Bangla literature using two machine learning approaches, Support Vector Machines (SVM) and Naive Bayes classifier. In the research they assembled Bengali Blogs scripted by twenty Bangladeshi authors of two different fields (Political, Educational). In the classification system, if five or six writers' documents are taken, then the model gives the best accuracy of 90.74% on SVM and 86.21% accuracy on Naive Bayes and the accuracy decreases to 73.64% on SVM and 70.38% on naive Bayes classification while taking 20 individual writers of both political and educational field. The proposal provides more accuracy than previously established works as all the collected corpus of the paper are of different writers writing, on the analogous field.

10. Shanta Phani et al (2016) [10] try to present an authorship attribution system for the blogs in the paper. They have presented a new Bengali blog corpus of 3000 passages written by three authors. In the paper, they proposed a text classification system, based on lexical features such as character bigrams and trigrams, word n-grams (n = 1, 2, 3) and stop words, using four classifiers and they achieve best results (more than 99%) on the held-out dataset using Multilayered Perceptron (MLP) amongst the four classifiers. They also claim that MLP provided better performance for both small and larger datasets.

Author / Year	Methodology	Dataset	Algorithm	Result	Contribution	Limitation
1. Hossain, A. S. et al. (2020) [1]	Author attribution through writing style analysis	Eight Bangladeshi bloggers' public blogs, including 1764 articles	Feedforward Neural Networks and Support Vector Machines	an average accuracy of 76.4% for selected features	Offers a potential solution for authorship detection	Limited to the analysis of Bangladeshi bloggers' writing style
2. Hossain, M. T. et al. (2017) [2]	Stylometric Analysis for Authorship Attribution in Bengali Literature	Writings of six Bangladeshi authors	Machine Learning Algorithm	90.67% accuracy when using five features	Introduced a flexible authorship attribution system	Limited variety of bloggers
3. Islam, M. A. et al. (2018) [3]	Author Attribution using Stylometry and Neural Network	1973 Bangla articles written by Six Different writers were used.	Neural Network Model, Multi Level Perceptron Model	85% accuracy for author identification		Small and Imbalanced Dataset. Interpretability , Scalability
4. Dhar, A. et al. (2022) [4]	Author Identification from Literary Articles	1200 literary articles written by 50 different authors	convolutional neural network (CNN)	maximum accuracy of 93.58%	Offer a better result for author identification on articles	Article dataset
5. Anisuzzaman et al. (2018) [5]	Authorship Attribution for Bengali Language	Bengali literature of three different authors	Fusion of N-Gram and Naive Bayes Algorithms	Naïve Bayes provides 86% accuracy and 95% by own technique	Offers an effective solution for authorship detection	Limited variety of authors
6. Aisha et al. (2019) [6]	Authorship Attribution using Character-Level CNN	News articles and online Bangla e-library containing writings of different authors	CNN with one-hot and pretrained embeddings	Around 85% on average. Accuracy grows with magnitude of dataset	Offers a time and memory efficient training process which is better than word embeddings	Performs poorly when dataset is small
7. Rajib et al. (2021) [7]	authorship classification approach	consisting of 949, 062 unlabeled	Support vector machine	CNN obtained the highest	develops new resources, evaluates 90	Limited generalizability, dependency

	using CNN	texts and 18 authors with 25, 749 labeled texts	(SVM), Naive Bayes (NB)	accuracy of 93.45% (for BACC-18), 95.02% (for BAAD16), and 98.67%(LD)	embedding models, optimizes 36 classification models, and achieves high accuracies.	on embedding techniques, reliance on accuracy metrics.
8. Khan, T. F. et al. (2023) [8]	Developed CNN architecture for authorship verification	Documents over 21,918, from newspaper websites.	CNN with hyper-tuned parameters, trained, and evaluated models on UAVC-22	CNN model verifies the authorship of Urdu text with a 93% accuracy rate	Created extensive Urdu corpus, introduced nine embedding models	computational complexity, and challenges with CNN decisions
9.Pal, U. et al. (2017) [9]	A Machine Learning Approach for Stylometric Analysis of Bangla Literature	Blogs scripted by twenty Bangladeshi authors of two different fields	Support Vector Machines (SVM) and Naive Bayes classifier	gives an accuracy of 73.64% on SVM and 70.38% on naive Bayes classification	provides more accuracy than previously established works mentioned in the paper	Accuracy decreases when the number of authors increases.
10.Shanta Phani et al. (2016) [10]	A Machine Learning Approach for Authorship Attribution for Bengali Blogs	Bengali blog corpus of 3000 passages written by three authors	Multilayer Perceptron (MLP)	get as high results as 99% for the system.	MLP gave better performance for larger datasets too.	The dataset is small.

Table 1: Summary of related works

3 Materials and Methods

3.1 Dataset

The Data was Collected from Mendeley. A dataset with sample Bangla texts from 16 authors containing a total of 13.4+ million words. The dataset was equally partitioned with each document having the same length of 750 words. It contains writings of 16 Bengali writers thus it was named BAAD16 [11]. This Dataset got 1882 views and 629 downloads on Mendeley and it has been used in many other research works.

3.2 Dataset Preprocessing

Dataset contains two columns:

1. Label (Author's name)
2. Text (Partitions collected from their writing)

From all the entries from the dataset 8,987 are accessed as per the data frame. The dataset was cleaned by dropping rows with null values.

We merged 2 rows where each contained ~750 words to a single row of ~1500 words to achieve better performance for our model.

The practice of correcting or eliminating inaccurate, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. There are numerous ways for data to be duplicated or incorrectly categorized when integrating different data sources. Even though results and algorithms seem correct, they are unreliable if the data is inaccurate. Because the procedures will differ depending on the dataset, there is no one set method that can be used to prescribe the precise steps in the data cleaning process. For this project, Initially, stop words, common words with limited semantic value, are removed to reduce noise and enhance clustering quality. Subsequently, the remaining text is transformed into numerical vectors using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, which assigns higher weights to terms that are more frequent within a document but less frequent across the entire dataset. This TF-IDF representation enables meaningful comparison and analysis of the textual data. Furthermore, dimensionality reduction techniques, particularly PCA [12][13] (Principal Component Analysis), are applied to reduce the high-dimensional TF-IDF vectors into a lower-dimensional space, facilitating visualization and computational efficiency in subsequent clustering tasks. Overall, this data cleaning process prepares the textual data for effective clustering analysis by converting it into a numerical format and reducing its dimensionality while preserving its underlying structure.

3.3 Design & Implementation

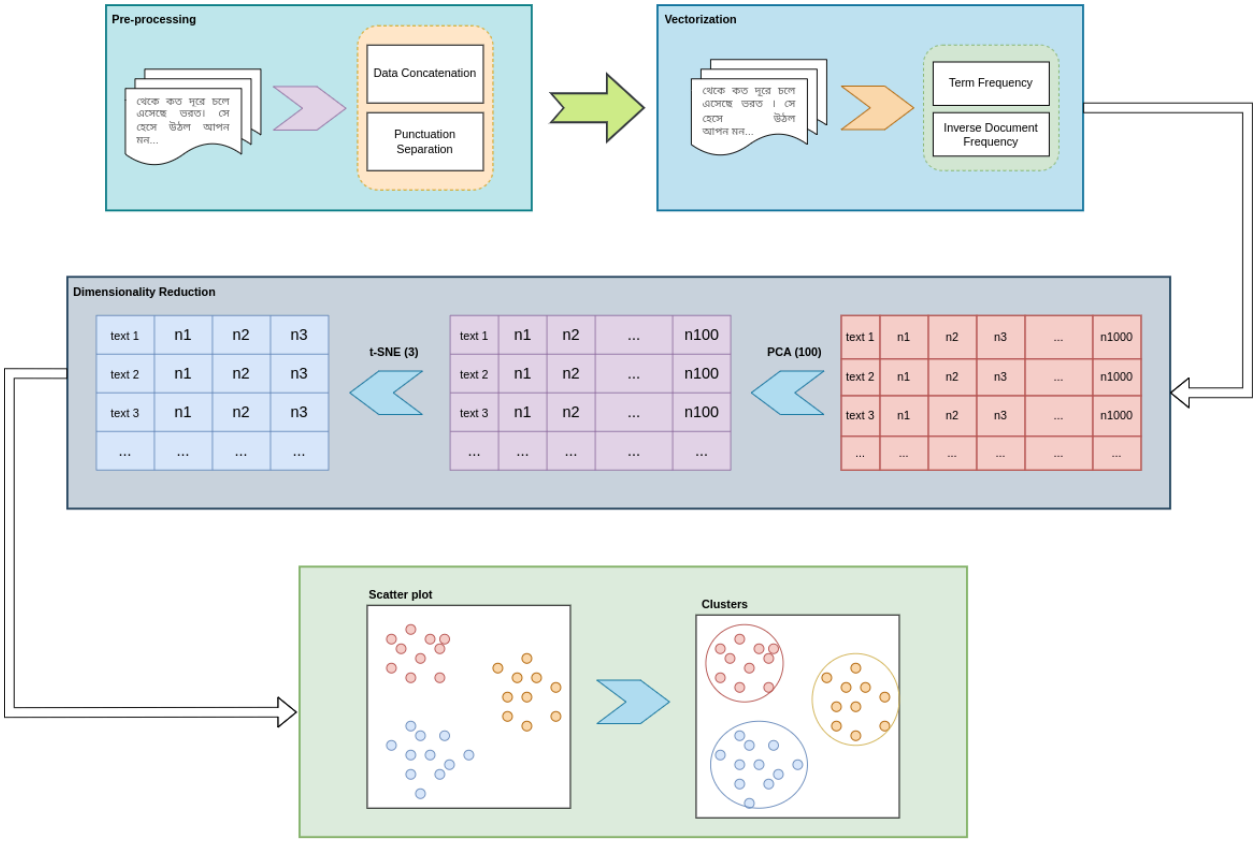


Figure 3: Overview of methodology

3.3.1 Overview

Our study aims to profile authors in Bangla text using advanced text clustering techniques. We developed a method that involves a series of steps, starting with data preprocessing, where rows are merged, and punctuation is separated from terms to prepare the text for analysis. We then create text embeddings (convert the text into numerical feature vectors) using TF-IDF, which captures term importance across the dataset.

To manage the high dimensionality of these vectors, we apply preliminary dimensionality reduction techniques such as Principal Component Analysis (PCA), and Latent Semantic Analysis (LSA) [14] to reduce the feature space, minimizing noise and enhancing the clarity of the clusters. Further dimensionality reduction is achieved using T-distributed Stochastic Neighbor Embedding (t-SNE) [15], allowing for effective visualization, clustering, and analysis.

For the clustering part, we used K-Means, and DBSCAN, two algorithms chosen for their complementary strengths in handling structured and unstructured data. The effectiveness of these

clustering methods is evaluated through a range of external and internal validation metrics, ensuring a thorough assessment of their performance in our study of author profiling.

3.3.2 Preprocessing

Data concatenation:

The BAAD16 dataset is a collection of written works(text), each paired with the author's identity(label). Each piece of writing contains 750 words. To gain a deeper understanding of the author's writing style, we merged consecutive pairs of same-author texts, creating new pieces with 1500 words each. This approach enhances the contextual richness of each text sample, enabling more precise stylometric analysis and more accurate clustering of similar authors.

Punctuation separation:

To make sure our analysis focuses on the important words and avoids any problems caused by punctuation, we separated punctuation marks from the main text. This also helps us understand how words and punctuation are related in a sentence. We developed a custom Bangla tokenizer function to do this for Bangla text.

This function splits each text entry using a regular expression that identifies both spaces and various punctuation symbols (such as commas, periods, exclamations, etc.). The regular expression treats punctuation as individual tokens while also removing any extra whitespace.

```
def tokenize_bangla(text: str):
 r =
re.compile(r'([\s\|{}]+)'.format(re.escape('!"#$%&\'()*+,’|-./:;<=>?@[\\]^_`{|}~')))
 list_: list[str] = r.split(text)
 list_ = [item.replace(" ", "").replace("\n", "").replace("\t", "") if not
item.isspace() else item for item in list_ if item.strip()]
 return list_
```

Figure 4: Custom Bangla tokenizer code snippet

The processed tokens are then recombined into a clean, punctuation-separated string, which is stored in a new column for subsequent analysis. By cleaning the text in this way, we ensure that punctuation does not interfere with the extraction of stylistic features, leading to more accurate vectorization, contextual analysis, and clustering.

3.3.3 Text Embeddings:

To transform the Bangla text data into a numerical format suitable for text analysis, we used the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer, a widely used technique in text mining and natural language processing.

TF-IDF is a statistical measure used to evaluate how important a word is to a document in a collection of documents. It is a combination of two factors - Term Frequency & Inverse Document Frequency.

Term Frequency (TF)

It measures how frequently a term appears in a particular document.

Formula:

$$TF(t, d) = \frac{Number\ of\ occurrences\ of\ term\ t\ in\ document\ d}{Total\ number\ of\ terms\ in\ document\ d}$$

The significance of Term Frequency is that the words that appear more frequently in a document are more likely to be relevant to its topic.

Inverse Document Frequency (IDF)

It measures how common or rare a term is across all documents in the collection. [16][17][19]

Formula:

$$IDF(t) = log(\frac{N}{Number\ of\ documents\ containing\ term\ t})$$

The Inverse Document Frequency annotates the words that appear in many documents as less informative than the words that only appear in a few documents.

Combining TF and IDF:

$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$

A word with a high TF-IDF score is both frequent in a document and rare in the collection. This indicates that the word is likely to be a strong indicator of the document's topic. On the other hand, a word with a low TF-IDF score is either infrequent in the document or common in the collection. This indicates that the word is less informative for the document's topic.

Hyper Parameter Configuration

For this study, the TF-IDF vectorizer was configured with the following settings:

Parameter	Value	Description
max_features	1000	Limits the number of features to the 1,000 most significant terms, focusing on the most relevant vocabulary and reducing computational load
ngram_range	(1, 2)	Considers both unigrams (single words) and bigrams (pairs of consecutive words), enabling the model to capture more complex patterns and stylistic nuances in the text
sublinear_tf	True	Applies logarithmic scaling to term frequencies, mitigating the influence of extremely frequent terms that could otherwise dominate the feature space [18]
max_df	0.85	Ignores terms that appear in more than 80% of the documents, as they are likely to be common words with limited discriminative power

Table 2: TF-IDF Vectorizer hyper-parameters

3.3.4 Dimensionality Reduction:

Dimensionality reduction is essential in managing the high-dimensional data produced by TF-IDF vectorization, ensuring that the clustering algorithms can effectively operate on the most relevant features. In this study, we implemented a two-step reduction process, combining Principal Component Analysis (PCA), or Latent Semantic Analysis (LSA) with t-SNE (t-Distributed Stochastic Neighbor Embedding). This approach strategically reduces noise, enhances cluster formation, and improves the overall efficiency and interpretability of the clustering results.

Initial Reduction (PCA/LSA)

The first step in the dimensionality reduction process involves condensing the high-dimensional feature space generated by the TF-IDF vectorization. Given the 1,000 features produced, applying Principal Component Analysis (PCA) or Latent Semantic Analysis (LSA) allows us to distill this feature space into a more manageable number of components—specifically, 100 principal components. This step is crucial as it removes noise and highlights the most informative features, smoothing the data and preparing it for more effective clustering.

PCA

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction while preserving as much variability as possible in the data. It does so by transforming the original variables into a new set of uncorrelated variables called principal components. These components are ordered such that the first few retain most of the variation present in the original variables.

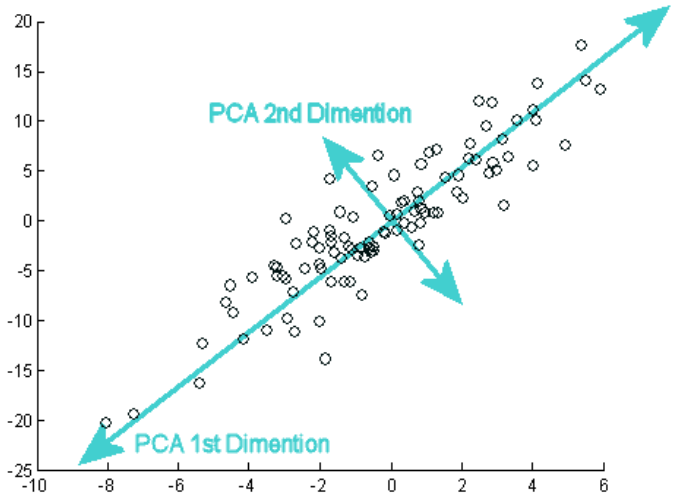


Figure 5: PCA process

LSA

Latent Semantic Analysis (LSA), also known as Latent Semantic Indexing (LSI), is a technique used to uncover the underlying semantic structure in text data. It is commonly used for reducing the dimensionality of text data by identifying patterns in the relationships between terms and documents. LSA is based on Singular Value Decomposition (SVD) and is particularly useful in identifying synonyms and polysemous words in documents.

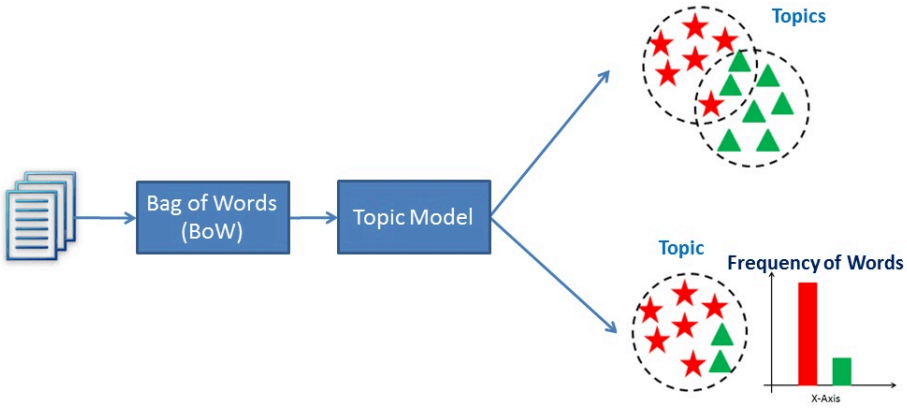


Figure 6: LSA process

Final Reduction (t-SNE)

After the preliminary reduction to 100 dimensions using PCA or LSA, t-SNE is applied to further reduce the data to 3 dimensions. This final reduction allows for clear visual representation of the clusters, aiding in the interpretation and validation of the clustering results.

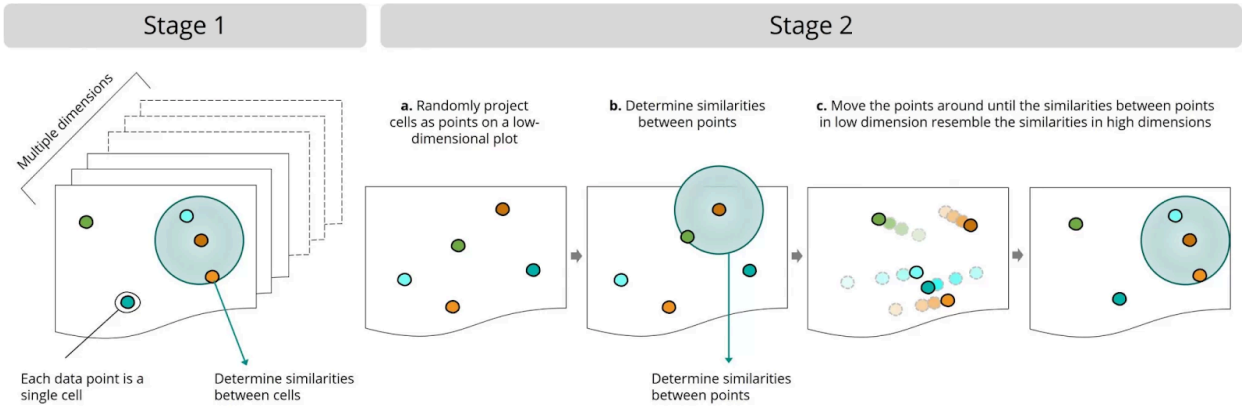


Figure 7: t-SNE steps

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a non-linear dimensionality reduction technique that specializes in visualizing high-dimensional data by focusing on preserving the local structure of the dataset. It works by converting similarities between data points into joint probabilities and then minimizing the divergence between these probabilities in the original and lower-dimensional spaces. This approach helps reveal complex patterns and relationships within the data, making t-SNE particularly useful for tasks such as clustering, exploratory data analysis, and generating intuitive visualizations. However, t-SNE can be computationally demanding and may struggle with very large datasets, requiring careful parameter tuning to balance between local and global structure preservation.

3.3.5 Clustering

K-means

K-Means is a widely used, iterative clustering algorithm that partitions data into *k* distinct clusters based on the similarity between data points. It aims to minimize the within-cluster variance by assigning each point to the cluster with the nearest mean (centroid). The simplicity, speed, and effectiveness of K-Means make it a popular choice for clustering tasks in high-dimensional spaces. [20]

In our study, K-Means clustering was applied to the low-dimensional data after the dimensionality reduction step (PCA/LSA followed by t-SNE). This allowed for an efficient clustering of the data based on writing styles or other latent features captured during preprocessing.

We chose k=16, which indicates the number of authors in our dataset.

Advantages of K-means:

- It provides a clear partitioning of the data into well-defined groups, making it easier to compare different clusters.
- It ensures consistency with classification models that require predefined groups, enabling a clearer evaluation of the clustering results in relation to external validation metrics like ARI and NMI.
- It is computationally efficient, especially for large datasets, making it suitable for high-dimensional text data after dimensionality reduction.
- It can handle large-scale datasets with relative ease, making it appropriate for your study, where clustering needs to be performed on reduced but still sizable data.

Challenges:
One potential challenge with K-Means is the need to specify the number of clusters *k* in advance. However, our choice of using *k* =16 aligns with the number of possible distinct author profiles we aim to capture, and this can be validated through external metrics.

DBSCAN

DBSCAN is a density-based clustering algorithm that groups data points based on their local density. It excels in identifying clusters of varying shapes and sizes while automatically detecting outliers, making it ideal for datasets where clusters are not well-separated or predefined. Unlike K-Means, DBSCAN does not require specifying the number of clusters in advance, which is one of its primary advantages in exploratory data analysis. [21]

DBSCAN works with three kind of data points:

1. Core Point: A point that has at least minPts neighbors within the distance ϵ. Core points form the internal structure of a cluster.
2. Border Point: A point that is not a core point but is within ϵ of a core point.
3. Noise Point (Outlier): A point that is neither a core nor a border point and doesn't belong to any cluster.

In our study, DBSCAN was used with great success after reducing the dimensionality of the dataset. DBSCAN is especially useful here because:

- Does not require specifying the number of clusters (*k*): This makes it more flexible than K-Means for cases where the number of clusters is not known a priori, or where the data naturally forms clusters of varying sizes.
- Handles noise and outliers: In our study of author profiling, there may be texts that don't conform to any particular writing style (such as outliers or noise). DBSCAN effectively identifies and isolates these points as noise, rather than forcing them into a cluster.
- Captures clusters of arbitrary shape: Since writing styles or linguistic patterns may not form spherical clusters (which is an assumption of K-Means), DBSCAN can handle irregularly shaped clusters by grouping points based on density rather than distance to a centroid.

Choosing ϵ and minPts:

Method	**ϵ**	**minPts**
PCA(3)	0.02	8
LSA(3)	0.0225	20
t-SNE(3)	2.5	8
PCA-then-t-SNE	2.5	8
LSA-then-t-SNE	2.5	8

Table 3: DBSCAN hyper parameters

Advantages of DBSCAN:

- No need to specify *k*: Unlike K-Means, DBSCAN automatically determines the number of clusters based on the density of the data.
- Outlier detection: It identifies noise points that don't fit well into any cluster, which is crucial for text clustering tasks where some documents may be anomalies.
- Flexibility: DBSCAN is capable of detecting clusters with different shapes, sizes, and densities, which is valuable for capturing nuanced patterns in writing styles.

Challenges of DBSCAN:

- Parameter sensitivity: DBSCAN's performance can be sensitive to the choice of ϵ and minPts. If ϵ is too small, many points will be labeled as noise, while if ϵ is too large, DBSCAN may merge distinct clusters.
- Difficulty with varying densities: DBSCAN struggles when clusters have significantly different densities, as a single value of ϵ may not capture the variations across clusters.

3.3.6 Evaluation metrics

We assessed the performance of the clustering algorithms using both external and internal evaluation metrics. These metrics allow us to objectively measure the quality of the clusters formed, ensuring the robustness and accuracy of our author profiling flow.

External validation metrics

External validation metrics compare the clustering results to external ground truth labels (in this case, the actual author identities from our dataset). These metrics provide insight into how well the clusters correspond to known categories.

Adjusted Rand Index (ARI): ARI measures the similarity between the clusters produced by the algorithm and the ground truth labels, adjusting for chance. It ranges from -1 (completely dissimilar) to 1 (perfect match). ARI is sensitive to the exact alignment of the clusters with the ground truth and rewards both correct cluster assignments and correct separation of different clusters.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_{i}\binom{a_i}{2}\sum_{j}\binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i}\binom{a_i}{2} + \sum_{j}\binom{b_j}{2}\right] - \left[\sum_{i}\binom{a_i}{2}\sum_{j}\binom{b_j}{2}\right]/\binom{n}{2}}$$

where n_{ij}, a_i, b_j are values from the contingency table.

$\binom{n}{2}$ is calculated as $n(n-1)/2$.

Normalized Mutual Information (NMI): NMI quantifies the amount of mutual information shared between the clusters and the ground truth labels, normalized to a range of 0 to 1. A value closer to 1 indicates that the clustering perfectly reflects the ground truth labels. NMI is particularly useful for assessing how well the clustering preserves relationships between instances.

$$NMI(Y, C) = \frac{2*I(Y;C)}{[H(Y)+H(C)]}$$

Adjusted Mutual Information (AMI): AMI is a variant of NMI that accounts for randomness by adjusting the mutual information score. Like NMI, it ranges from 0 to 1, with 1 indicating a perfect match with the ground truth.

$$AMI(U, V) = \frac{MI(U,V)-E\{MI(U,V)\}}{max\{H(U),H(V)\}-E\{MI(U,V)\}}$$

Fowlkes-Mallows Index (FMI): FMI evaluates the similarity between the clusters and the ground truth by considering precision and recall. It is the geometric mean of the precision and recall values. FMI values closer to 1 suggest a high degree of similarity between the clusters and the true labels.

$$FMI = \sqrt{PPV.TPR} = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}$$

where,

TP is the number of pairs of points that are present in the same cluster in both A_1 and A_2.

FP is the number of pairs of points that are present in the same cluster in A_1 but not in A_2.

FN is the number of pairs of points that are present in the same cluster in A_2 but not in A_1.

TN is the number of pairs of points that are in different clusters in both A_1 and A_2.

Internal validation metrics

Internal validation metrics assess the compactness and separation of the clusters without reference to external labels. These metrics provide insight into how well-defined the clusters are based on the inherent structure of the data.

Homogeneity: Homogeneity measures how uniform the clusters are, with higher values indicating that each cluster contains data points belonging to a single class. It assesses whether the clusters contain instances from only one true author.

Completeness: Completeness complements homogeneity by evaluating whether all points from the same true class (author) are assigned to the same cluster. High completeness values suggest that instances of the same class are not split across multiple clusters.

V-Measure: V-Measure is the harmonic mean of homogeneity and completeness, providing a balanced assessment of both criteria. A higher V-Measure indicates that the clustering is both consistent within clusters and does not fragment true classes across multiple clusters. V-Measure is identical to the NMI score with the arithmetic averaging method.

3.4 Project Management Features

3.4.1 Business Model Canvas

This section has explained the project management process of the capstone project. The Prototype named styloverse was developed with all the project management concepts to reflect real-life project development. Styloverse was built to expand our understanding of deployment level project management. Developing required documentation for a project has helped us gain essential knowledge about the capstone project's budgeting, resource allocation, and work breakdown structure. Developing Styloverse from an entrepreneurial point of view has enlarged our understanding of focusing on the work that matters, free from the distractions caused by tasks going off track or budgets spinning out of control. Considering it as a professional project. It has required around 4-5 months to build this web-based application except for the Research and literature study part. The Business model that was followed during this timespan is shown in figure 7.

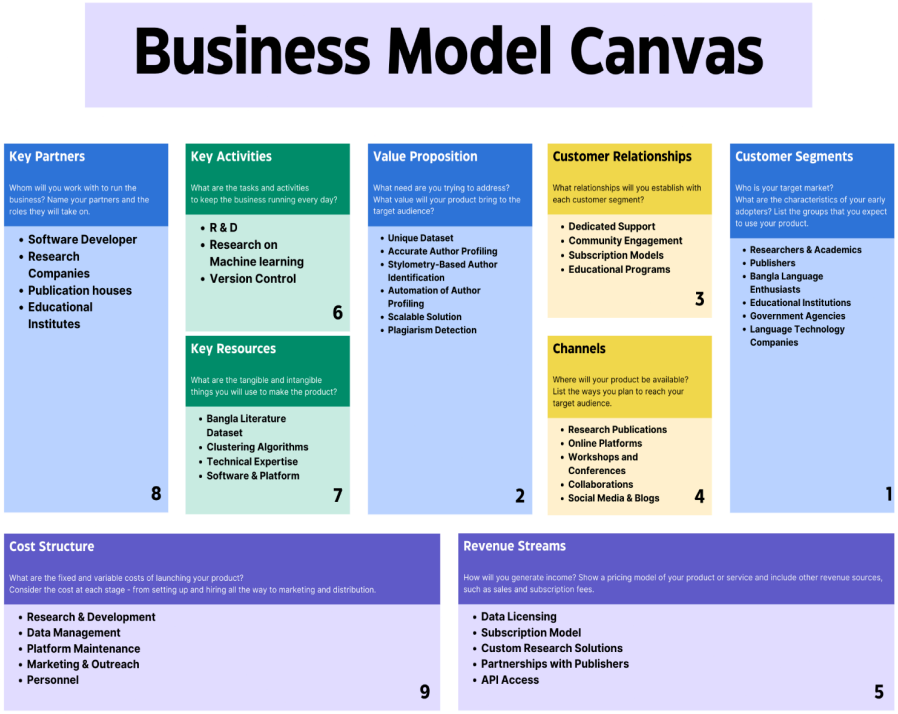


Figure 8: Business model canvas

3.4.2 Work Breakdown Structure

The Business Model Canvas (BMC) illustrated in Figure 7 serves as a crucial tool for project management by breaking down nine essential components of the business into a clear and

organized structure. These components provide a framework to visualize how different elements of the project interact and contribute to overall success. By using the BMC, the project team can identify potential gaps or misalignments between these components, which aids in refining strategies and improving planning.

The prototype's BMC is particularly useful in highlighting its core value propositions—the unique benefits and solutions the project offers to its target customers. It defines the specific offerings of the prototype, ensuring that its value is communicated effectively to the intended audience. Additionally, the BMC sheds light on the project's target audience and the relationships the project intends to build with its customers, whether through personalized services, automated support, or other channels.

Another critical aspect covered in the BMC is the strategic approach the project will take to reach its customers, including the channels and platforms that will be used for engagement and outreach. This helps in planning how the project will connect with customers and deliver its value proposition effectively.

Moreover, the BMC outlines the project’s key partners—individuals, organizations, or stakeholders essential for development and operational success. It also lays out the project's cost structure, identifying key expenses, and the revenue model, showing how the project plans to generate income and sustain itself financially. Together, these elements provide a comprehensive overview of the project's business strategy, ensuring all aspects are aligned and reinforcing each other for effective execution.

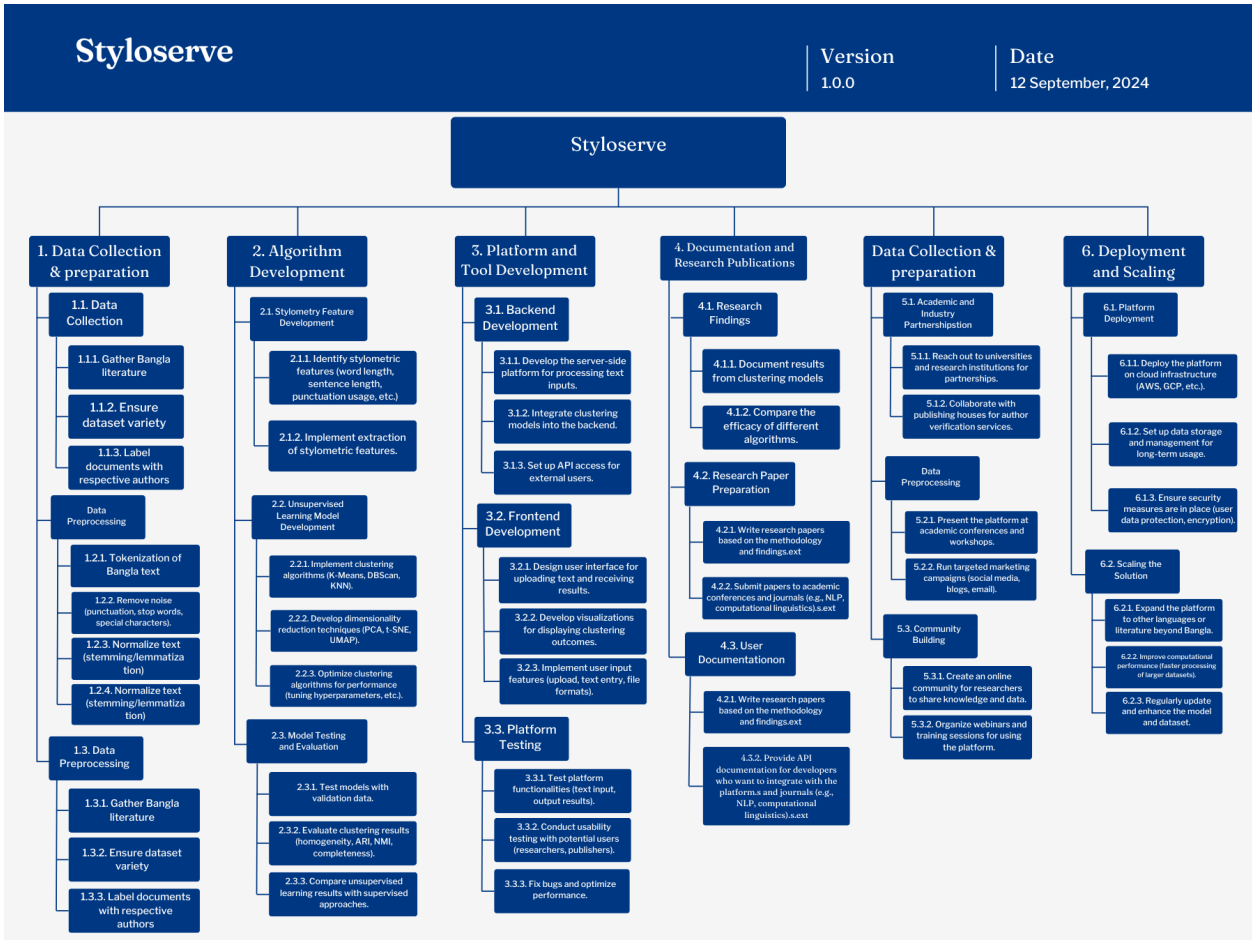


Figure 9: Work Breakdown Structure

3.4.3 Resource Allocation

Task/Phase	Team Members	Time Allocated	Tools/Resources Needed	Estimated Cost (Dummy)
1. Data Collection and Preparation				
1.1 Dataset Collection	Data Scientist, Intern	2 weeks	Digital libraries, data scraping tools	$2,000
1.2 Data Preprocessing	Data Scientist, NLP Expert	3 weeks	Python, NLTK, spaCy	$3,500
1.3 Data Validation	Data Scientist	1 week	Python, Pandas	$1,500

2. Algorithm Development				
2.1 Stylometry Feature Development	NLP Expert, Data Scientist	4 weeks	Python, custom feature extraction scripts	$4,500
2.2 Clustering Algorithm Development	Data Scientist, ML Expert	6 weeks	Scikit-learn, K-Means, DBSCAN, KNN	$7,000
2.3 Model Testing and Evaluation	ML Expert, Researcher	3 weeks	Scikit-learn, evaluation scripts	$3,500
3. Platform and Tool Development				
3.1 Backend Development	Full Stack Developer	5 weeks	Python (Flask/Django), REST API	$6,000
3.2 Frontend Development	Frontend Developer	4 weeks	HTML/CSS, JavaScript, React.js, Three.js	$4,000
3.3 Platform Testing	Full Stack Developer, QA	2 weeks	Selenium, Postman, manual testing	$2,500
4. Documentation and Research				
4.1 Research Paper Preparation	Researcher, Technical Writer	4 weeks	LaTeX, Google Docs	$3,000
4.2 User Documentation	Technical Writer	3 weeks	Markdown, GitHub Pages, API documentation tools	$2,000
5. Marketing and Outreach				
5.1 Academic and Industry Partnerships	Marketing Lead, Researcher	6 weeks	Email campaigns, online marketing tools	$5,000

5.2 Promotion (Conferences, Webinars)	Marketing Lead, Presenter	Ongoing (post-launch)	Zoom, Webinar platforms, conference tickets	$3,500 (initial)
5.3 Community Building	Marketing Lead, Researcher	Ongoing (post-launch)	Social media tools, community forums	$1,500 (initial)
6. Deployment and Scaling				
6.1 Platform Deployment	DevOps Engineer	3 weeks	AWS/GCP, cloud storage, security tools	$5,000
6.2 Scaling the Solution	DevOps Engineer, ML Expert	Ongoing	Server upgrade, cloud optimization	$3,000 (monthly)
Total Estimated Cost				**$58,500 + $3,000/month**

Table 4: Resource allocation

3.4.4 Critical Path Method

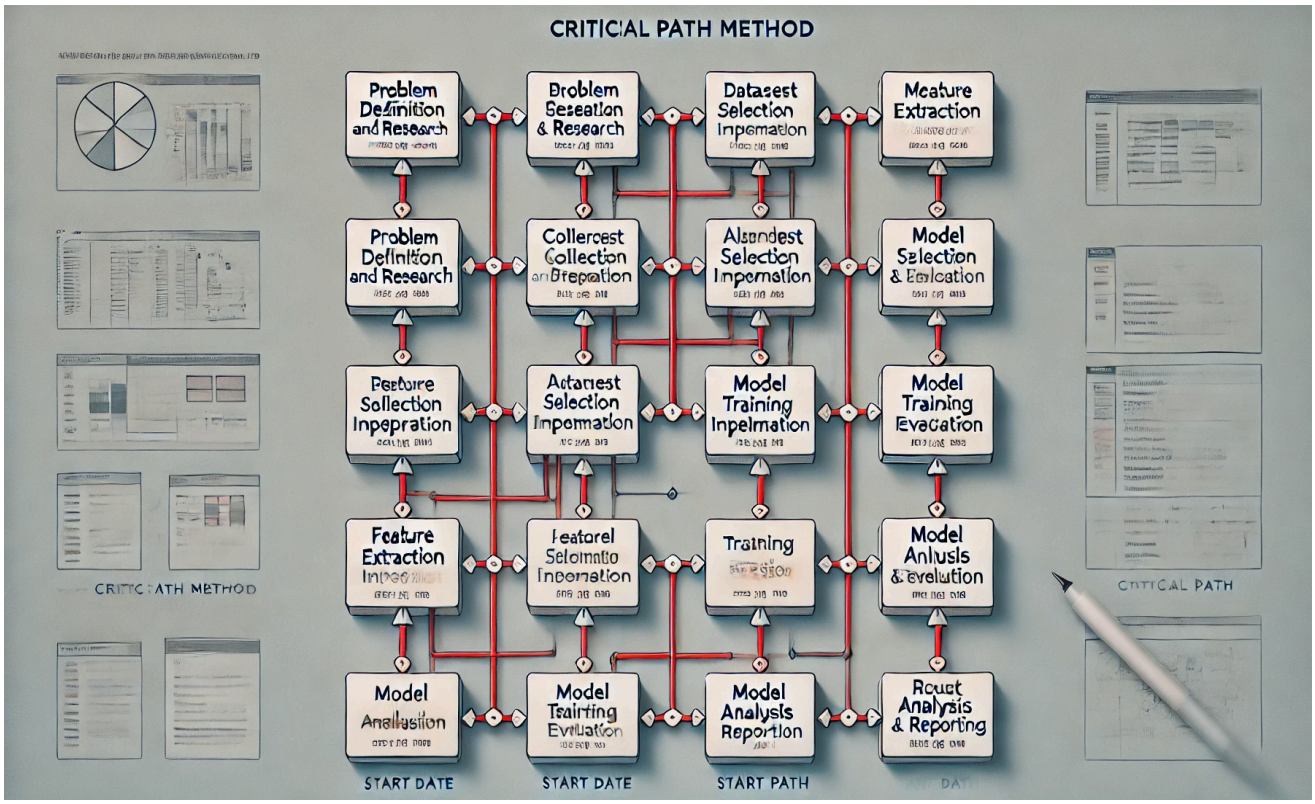


Figure 10: Critical Path Method

3.4.5 Infrastructure Cost

Category	Estimated Cost (monthly)
Cloud Storage	$20–$100
GPU Compute (Training)	$300–$800
Compute Instances (Non-GPU)	$100–$300
Software/Cloud Deployment	$200–$400
Data Scientists/ML Engineers	$10,000–$30,000
Frontend/Backend Developers	$8,000–$16,000
Project Manager	$5,000–$9,000
Data Acquisition	$500–$1,000 (one-time)
Miscellaneous (Support)	$200–$500

Table 5: Infrastructure cost

Total Monthly Costs:

- Minimum: ~$24,320/month
- Maximum: ~$58,100/month

3.4.6 Break-Even Point Calculation

To calculate the break-even point (BEP) for your project, we need to assess the fixed costs, variable costs, and revenue from the product or service. The break-even point occurs when total revenue equals total costs, meaning no profit or loss is made.

Key Formula for Break-Even Point (BEP):

$$\text{BEP} = \frac{\text{Fixed Costs}}{\text{Revenue per Unit} - \text{Variable Cost per Unit}}$$

Where:

- Fixed Costs are the ongoing expenses that do not change with production volume (e.g., infrastructure, salaries).
- Variable Costs are costs that vary with each unit produced (e.g., cost per customer served, maintenance, etc.).

- Revenue per Unit is how much you earn for each unit sold.

Step-by-Step Break-Even Calculation Example:

1. Fixed Costs (Monthly):

These are costs that remain constant regardless of the number of clients or units sold:

- **Infrastructure (cloud, compute, etc.)**: $2,000/month
- **Salaries (team costs)**: $25,000/month
- **Software and tools**: $500/month
- **Miscellaneous (maintenance, support)**: $300/month

Total Fixed Costs = **$27,800/month**

2. Variable Costs (Per Customer):

These are the costs incurred per customer served:

- **Data processing and compute usage**: $100/customer
- **Customer support and services**: $50/customer

Total Variable Costs per customer = **$150/customer**

3. Revenue Per Unit (Per Customer):

Let's assume the product (Bangla Author Profiling service) is sold at:

- **Revenue per customer**: $500/customer

Break-Even Point Calculation:

Using the formula:

$$\text{BEP} = \frac{\text{Fixed Costs}}{\text{Revenue per Unit} - \text{Variable Cost per Unit}}$$

$$\text{BEP} = \frac{27,800}{500 - 150} = \frac{27,800}{350} = 79.43$$

Thus, you need to serve **80 customers per month** to break even.

Break-Even Summary:

- **Fixed Costs**: $27,800/month
- **Variable Costs per customer**: $150
- **Revenue per customer**: $500
- **Break-Even Point**: You need to serve **80 customers/month** to cover your costs and reach the break-even point.

This gives you a clear idea of the customer volume required to cover your costs. If you acquire more than 80 customers in a month, you will start making a profit.

3.5 Materials and Devices

3.5.1 Materials

Dataset:
We used BAAD16: Bangla Authorship Attribution Dataset for this study. It is a dataset with sample Bangla texts from 16 authors containing a total of 13.4+ million words. The dataset was equally partitioned with each document having the same length of 750 words.

Software & Libraries:
The analysis was conducted using Python (version 3.10.12).
Key libraries used include:
scikit-learn (version 1.3.2) for vectorization (TfidfVectorizer), dimensionality reduction (PCA, LSA, and t-SNE), clustering algorithms (K-Means and DBSCAN), and evaluation metrics.
Pandas (version 2.1.4) for data manipulation and preprocessing.
Matplotlib for generating visualizations of clusters and evaluation metrics.

Environment:
The programming for our research was executed using Google Colab, an online integrated development environment (IDE) which provides a Jupyter Notebook that facilitates interactive exploration and visualization of data. Google Colab provided access to cloud-based resources, including a GPU for efficient computation.

3.5.2 Devices

Hardware:
CPU: 1x single core hyper threaded Xeon Processors @2.3Ghz (1 core, 2 threads)
RAM: 13 GB
GPU: Tesla T4 GPU (for acceleration of t-SNE computations)
Storage: 108 GB

4 Results

We compared the impact of different dimensionality reduction techniques on the performance of the clustering algorithms. We evaluate the clustering metrics without dimensionality reduction, with PCA or LSA, and with t-SNE, as well as combinations of PCA or LSA followed by t-SNE. This allowed us to assess how each reduction method influences the quality of the clusters.

4.1 Without dimensionality reduction

Without any dimensionality reduction, the algorithms clustered the full 1000-dimensional feature set directly. As shown in Table 1, the clustering results, though decent, indicate a certain degree of noise and overlap between the clusters. Although we could see evaluation metrics for the K-Means algorithm, the DBSCAN algorithm could not provide us with any result. Which suggests difficulty in capturing the inherent structure of the data in such a high-dimensional space.

Table 6: Metrics Without Dimensionality Reduction

Algorithm	**ARI [-1, 1]**	**NMI**	**AMI**	**FMI**	**Homogeneity**	**Completeness**	**V-measure**
K-Means	0.66170	**0.85541**	0.85471	0.70935	**0.89559**	0.81869	0.85541
DBSCAN	0	0	0	**0.33227**	0	**1.0**	0

4.2 With PCA or LSA to 3 Components

Reducing the features to 3 components using PCA and LSA decreased the clustering performance compared to no dimensionality reduction for K-means. But for the DBSCAN algorithm, reducing dimensions gave a good result. PCA, which focuses on preserving variance, provided a slightly better separation of clusters, as indicated by higher ARI and NMI scores. LSA, which is a matrix factorization technique, also performed well but with more focus on the underlying semantic structure.

Table 7: Metrics PCA to 3 components

Algorithm	**ARI**	**NMI**	**AMI**	**FMI**	**Homogeneity**	**Completeness**	**V-measure**
K-Means	0.45447	**0.62181**	0.61992	0.51175	**0.64649**	0.60333	0.62181
DBSCAN	0.25245	**0.44945**	0.44748	0.40268	0.35575	**0.61249**	0.44945

Table 8: Metrics LSA to 3 components

Algorithm	ARI	NMI	AMI	FMI	Homogeneity	Completeness	V-measure
K-Means	0.33873	**0.55132**	0.54912	0.40585	**0.57788**	0.52751	0.55132
DBSCAN	0.38568	0.50018	0.49928	**0.53643**	0.38425	**0.71678**	0.50018

Although the validation metrics are low, both PCA and LSA gave a good insight for a 3D visualization of the feature space.

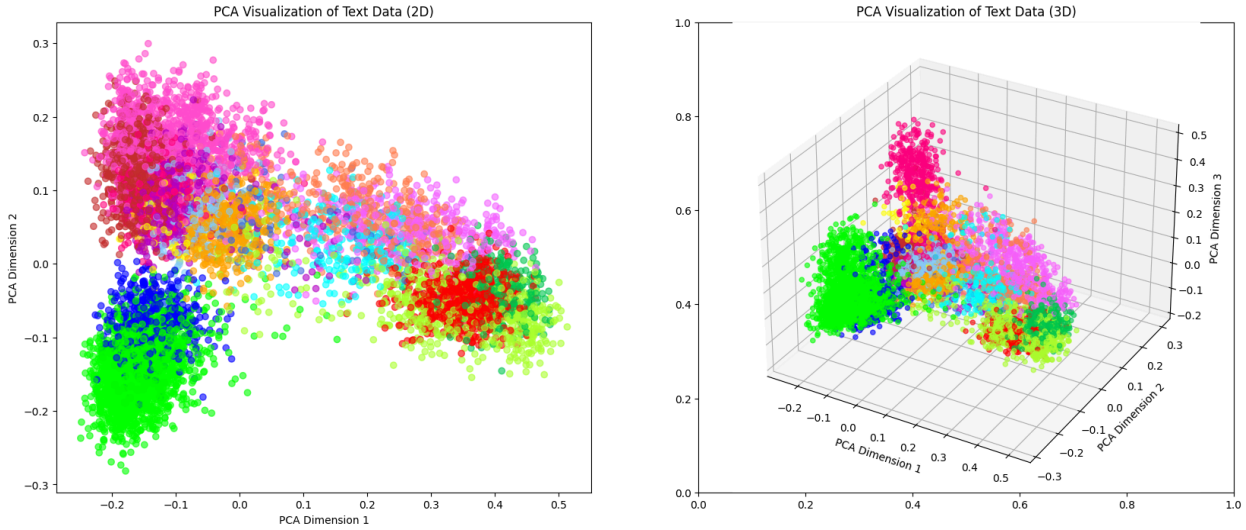


Figure 11: PCA(n_components=3) visualization for 2D & 3D

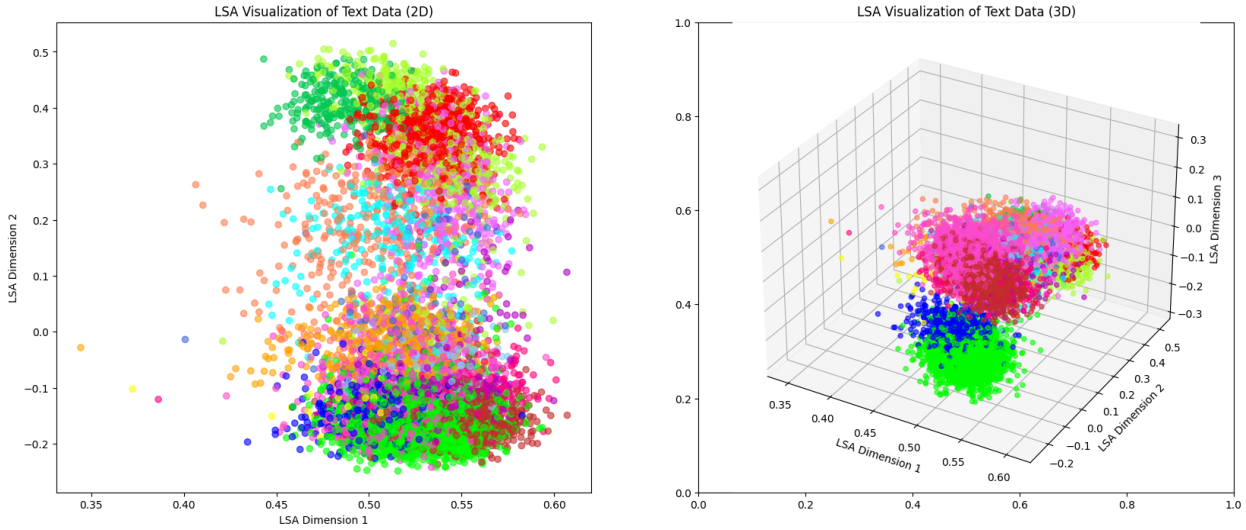


Figure 12: LSA(n_components=3) visualization for 2D & 3D

4.3 With t-SNE to 3 Components

Applying t-SNE to reduce the data to 3 components produced highly distinctive clusters, as t-SNE is designed to optimize local neighborhood relationships in the data. This method resulted in improved clustering performance for both K-Means and DBSCAN, especially in terms of homogeneity and completeness. However, t-SNE is computationally more expensive and sensitive to hyperparameters - which requires careful tuning.

Table 9: Metrics t-SNE to 3 components

Algorithm	**ARI**	**NMI**	**AMI**	**FMI**	**Homogeneity**	**Completeness**	**V-measure**
K-Means	0.61029	**0.83014**	0.82931	0.66468	**0.87428**	0.79025	0.83014
DBSCAN	0.53774	**0.78408**	0.77853	0.58662	**0.84122**	0.73421	0.78408

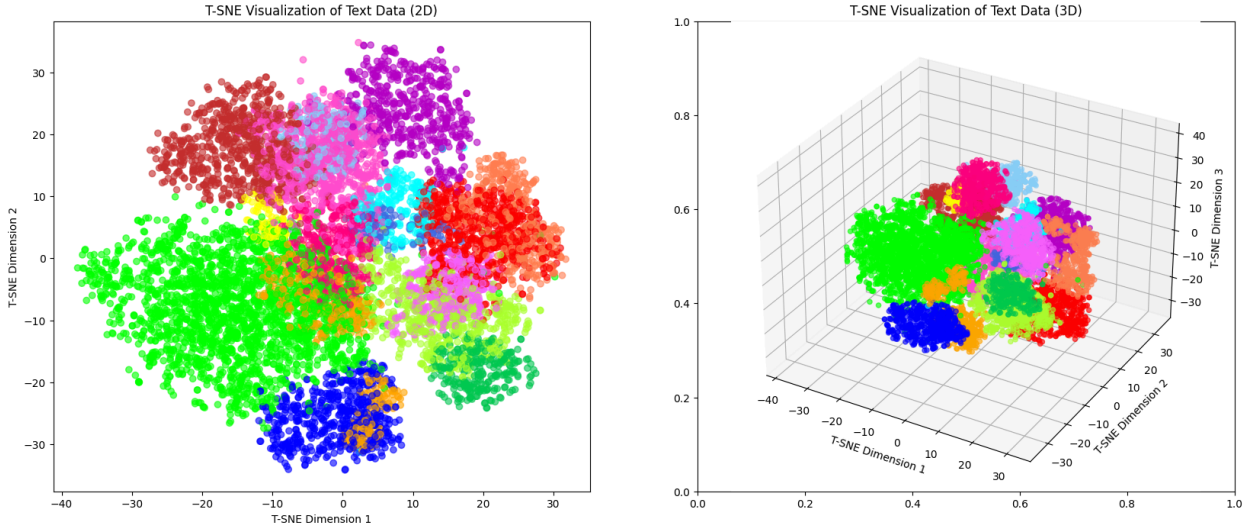


Figure 13: t-SNE(n_components=3) visualization for 2D & 3D

4.4 With PCA then t-SNE

Combining PCA with t-SNE for dimensionality reduction offered a balance between preserving global variance and optimizing local neighborhoods. By first reducing the data to 100 components with PCA, the computational load on t-SNE was reduced, leading to faster convergence and better-defined clusters. This combination produced higher scores in external validation metrics for DBSCAN algorithm and a decent score for K-means algorithm. The internal validation metrics were also up to par.

Table 10: Metrics PCA(100) then t-SNE(3)

Algorithm	ARI	NMI	AMI	FMI	Homogeneity	Completeness	V-measure
K-Means	0.65450	**0.87165**	0.87102	0.70460	**0.91514**	0.83210	0.87165
DBSCAN	0.93861	**0.94618**	0.94569	0.94538	**0.95811**	0.93455	0.94618

This approach gave a decent distinction between clusters when visualizing in 2D & 3D space.

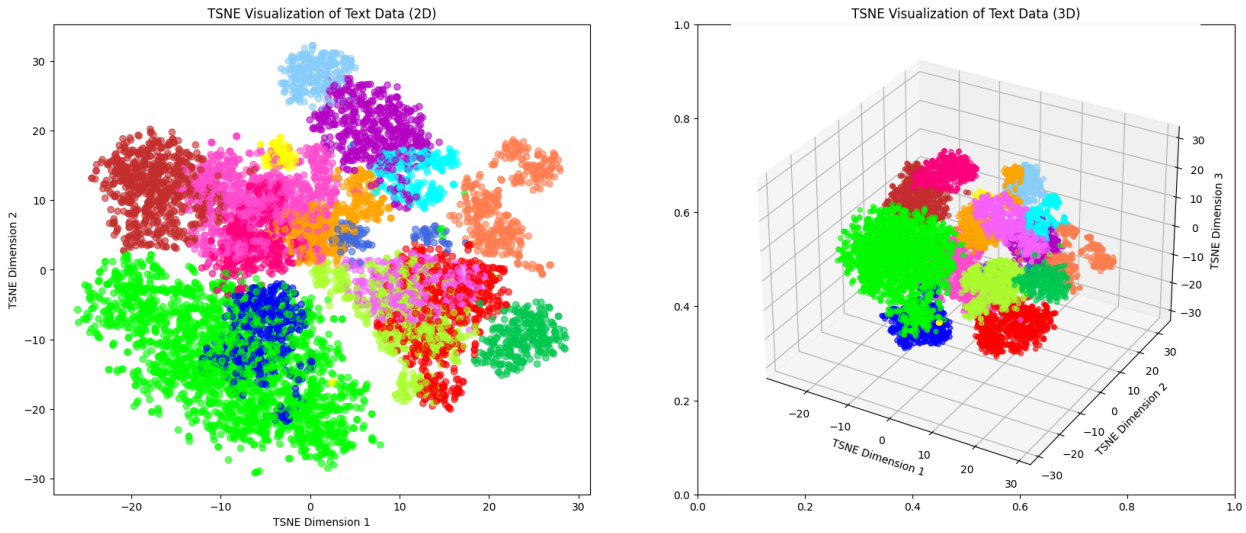


Figure 14: PCA(3) then t-SNE(3) visualization for 2D & 3D

4.5 With LSA then t-SNE

Using LSA to reduce dimensionality before applying t-SNE provided a little different clustering results compared to PCA. LSA focuses on capturing the latent semantic relationships, which enhanced the clustering performance in DBSCAN, as shown by its higher homogeneity and completeness scores. However, the K-Means algorithm slightly underperformed compared to the PCA-then-t-SNE pipeline.

Table 11: Metrics LSA(100) then t-SNE(3)

Algorithm	ARI	NMI	AMI	FMI	Homogeneity	Completeness	V-measure
K-Means	0.64324	**0.85476**	0.85405	0.69437	**0.89801**	0.81549	0.85476
DBSCAN	0.85289	**0.90765**	0.90678	0.87074	0.90106	**0.91433**	0.90765

This approach also gave an adequate distinction between clusters when visualizing in 2D & 3D space.

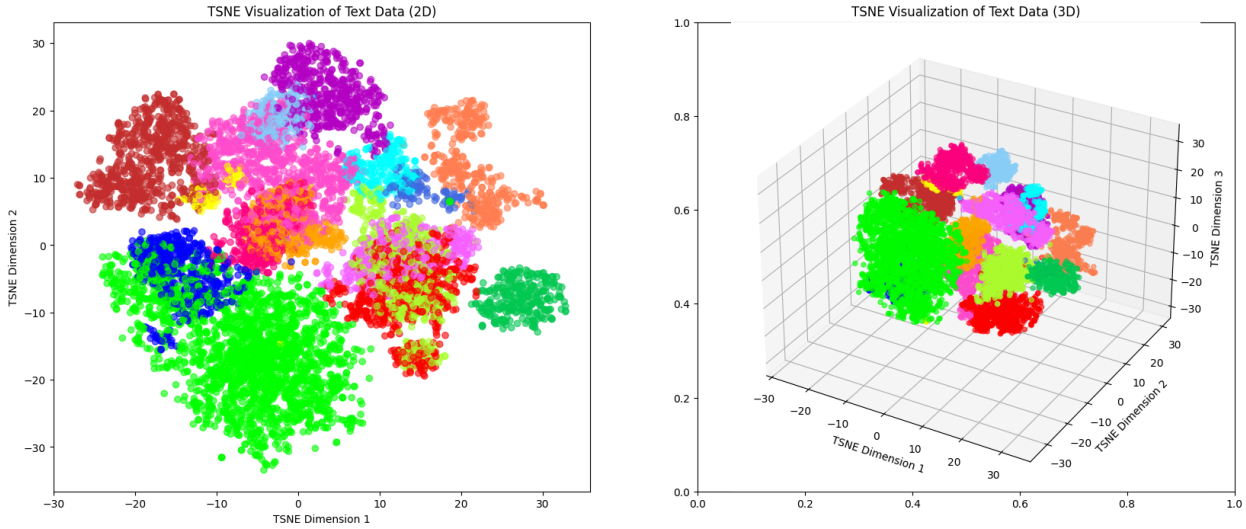


Figure 15: LSA(3) then t-SNE(3) visualization for 2D & 3D

In summary, our experiments show that applying dimensionality reduction significantly improves clustering performance. t-SNE, particularly when combined with PCA or LSA, produced the most distinct clusters, with PCA-then-t-SNE offering the best overall performance for K-Means, while LSA-then-t-SNE enhanced the performance of DBSCAN. These results suggest that different reduction methods and their combinations offer unique advantages depending on the clustering technique used.

5 Conclusion & Future work

In our research, we applied stylometry-based author attributionship on Bangla text using text clustering techniques. We explored the performance of clustering algorithms like K-Means and DBSCAN, both with and without dimensionality reduction. Our results demonstrated that dimensionality reduction significantly improves cluster performance, especially when using a 2-step process like PCA followed by t-SNE. The clustering algorithms, when combined with the right reduction techniques, yielded improved metrics such as ARI, NMI, and homogeneity, which reflects more distinct and meaningful clusters.

Among the dimensionality reduction methods we explored, the combination of PCA and t-SNE proved to be the most effective for both K-Means and DBSCAN. This suggests that combining global and local data structures through hybrid dimensionality reduction approaches helps uncover latent patterns in high-dimensional data.

Although we acquired adequate results, our study had certain limitations. t-SNE is computationally expensive and sensitive to hyperparameter tuning, which adds complexity to the analysis process. Additionally, while the results are promising for the dataset used, further evaluation is required to ensure the generalizability of the findings across other Bangla text corpusses.

6 Reference

[1] Hossain, A. S., Akter, N., & Islam, M. S. (2020). A stylometric approach for author attribution systems using neural network and machine learning classifiers. In ICCA 2020: Proceedings of the International Conference on Computing Advancements. Association for Computing Machinery. https://doi.org/10.1145/3377049.3377079

[2] Hossain, M. T., Rahman, M. M., Ismail, S., & Islam, M. S. (2017). A Stylometric Analysis on Bengali Literature for Authorship Attribution. In 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE. https://doi.org/10.1109/iccitechn.2017.8281768

[3] Islam, M. A., Kabir, M. M., Islam, M. S., & Tasnim, A. (2018). Authorship Attribution on Bengali literature using stylometric features and neural network. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT). IEEE. https://doi.org/10.1109/ceeict.2018.8628106

[4] Dhar, A., Mukherjee, H., Sen, S., Sk, M. O., Biswas, A., Gonçalves, T., & Roy, K. (2022). Author Identification from Literary Articles with Visual Features: A Case Study with Bangla Documents. Future Internet, 14(10), 272. https://doi.org/10.3390/fi14100272

[5] Anisuzzaman, D. M., & Salam, A. (2018). Authorship attribution for Bengali language using the fusion of N-Gram and Naive Bayes algorithms. International Journal of Information Technology and Computer Science, 10(10), 11–21. https://doi.org/10.5815/ijitcs.2018.10.02

[6] Khatun, A., Rahman, A., Islam, M. S., & E Jannat, M. (2019). Authorship attribution in Bangla literature using character-level CNN. In 2019 22nd International Conference on

Computer and Information Technology (ICCIT). IEEE. https://doi.org/10.1109/iccit48885.2019.9038560

[7] Hossain, M. R., Hoque, M. M., Dewan, M. a. A., Siddique, N., Islam, M. N., & Sarker, I. H. (2021b). Authorship classification in a resource constraint language using convolutional neural networks. IEEE Access, 9, 100319–100338. https://doi.org/10.1109/access.2021.3095967

[8] Khan, T. F., Anwar, W., Arshad, H., & Abbas, S. N. (2023). An empirical study on authorship verification for low resource language using Hyper-Tuned CNN approach. IEEE Access, 11, 80403–80415. https://doi.org/10.1109/access.2023.3299565

[9] Pal, U., Nipu, A. S., & Ismail, S. (2017). A machine learning approach for stylometric analysis of Bangla Literature. In 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE. https://doi.org/10.1109/iccitechn.2017.8281800

[10] Phani, S., Lahiri, S., & Biswas, A. (2016). A machine learning approach for authorship attribution for Bengali blogs. In 2016 International Conference on Asian Language Processing (IALP). IEEE. https://doi.org/10.1109/ialp.2016.7875984

[11] Khatun, Aisha; Rahman, Anisur; Islam, Md. Saiful (2020), "BAAD16: Bangla Authorship Attribution Dataset", Mendeley Data, V4, https://doi.org/10.17632/6d9jrkgtvv.4

[12] Pearson, K. (1901) On Lines and Planes of Closest Fit to Systems of Points in Space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2, 559-572. https://doi.org/10.1080/14786440109462720

[13] Jolliffe, I. (2002). Principal component analysis. Springer Science & Business Media.

[14] Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Fumas and L. L. Beck. "Improving information retrieval using latent semantic indexing." (1988).

[15] Van der Maaten, L. J. P., & Geoffrey, H. (2008). Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research, 9, 2579–2605. https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

[16] Jones, K. S. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. Journal of Documentation, 28(1), 11–21. https://doi.org/10.1108/eb026526

[17] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Scoring, term weighting, and the vector space model. In Cambridge University Press eBooks (pp. 100–123). https://doi.org/10.1017/cbo9780511809071.007

[18] TFIDF statistics | SAX-VSM. (n.d.). https://jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html

[19] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. Journal of Documentation, 60(5), 503–520. https://doi.org/10.1108/00220410410560582

[20] Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129–137. https://doi.org/10.1109/tit.1982.1056489

[21] Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial Databases with Noise. Knowledge Discovery and Data Mining, 226–231. https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

7 Appendix

7.1 Appendix A

Unsupervised machine learning and Clustering Algorithms

CO	**CO Description**	**K**	**EP/EA**	**PO**
CO1	We have gathered knowledge from Unsupervised machine learning and Clustering Algorithms in order to find a real-life complex problem as our capstone project which is Author Detection. Various problems will be solved by solving this problem.	(i) Identify a real-life problem [K1, K2, K3, K4] K1: Theory-based natural Sciences: K1: Theory-based natural Sciences: Our work is about Unsupervised machine learning and Clustering Algorithms. So, we have gained both theoretical and practical knowledge from previously done courses. We need in-depth knowledge about linear algebra, distance metrics, optimization, and probability K2: Conceptually-based mathematics, numerical analysis, statistics, and formal aspects of computer and information science: We have selected our capstone project based-on Unsupervised machine learning and Clustering Algorithms. So, mathematics, statistical analysis and data science are common concepts to be used. These are all based on	(i) Identify a real-life problem [EP1, EP2, EP3, EP4, EP5, EP6, EP7] EP1: Depth of knowledge required: In this project, we tried to apply our knowledge of engineering, design and practice. EP2: Range of conflicting requirements: We include extended engineering, conflicting technical and other challenges such as different types of models and data augmentation approaches in our system. EP3: Depth of analysis required: There is no particular solution to the problem. So, we have implemented and tried several approaches to find the best solution for our problem.	

		numerical analysis. K3: Theory-based engineering fundamentals: We need the knowledge of programming language to build our project. As we are using Unsupervised machine learning and Clustering Algorithms, we need a clear understanding of a programming language and its fundamentals. In order to finish our project perfectly, we need development knowledge as well K4: Forefront engineering specialist knowledge for practice: We have forefront engineering specialist knowledge of python, Unsupervised machine learning.	EP4: Familiarity of issues: It was a bit hard to collect a large amount of dataset. But we have comfortably collected a decent amount of data. EP5: Extent of applicable codes: We have used Unsupervised machine learning and Clustering Algorithms to solve this problem. The Engineering standard is maintained. To build this project, coding knowledge, implementation knowledge has been used. EP6: Extent of stakeholder involvement and conflicting requirements: The point of view of stakeholders for this project is considered. EP7: Interdependence: We have worked on high-level problems with linguistic richness and polysemy to solve this problem.	
CO2	As the detection of authors from	(i) Define the problems [K8] K8: Research	(i) Define the	

	classical Bangla literature continues to be a growing concern, our project aims to provide a solution that can accurately detect and flag such content. By identifying and addressing these instances early on, we hope to promote a safer and more respectful online community	Literature: We reviewed research papers related to Author Attribution and detection as well as software tools that can be used in our project. We are confident that this knowledge will help us effectively tackle the issue.	problems [EP1, EP2, EP3, EP4, EP5, EP6, EP7] [Same as (CO1)]	

CO	**CO Description**	**K**	**EP/EA**	**PO**
CO3	Analyze various aspects of the objectives for designing a solution for the capstone project.	**K2: Conceptual Knowledge** – Understand how to generate text embeddings using TF-IDF and its importance in clustering analysis. **K3: Procedural Knowledge** – Apply K-Means and DBSCAN clustering algorithms to group similar data points based on text embeddings.	**EP 1** Demonstrating knowledge of key clustering concepts such as K-Means, DBSCAN, and text embeddings like TF-IDF. **EP 2** Applying clustering algorithms to real-world data and profiling authors based on their text using unsupervised methods. **EP 3** Analyzing different clustering outcomes by using validation metrics and understanding why certain algorithms work better in specific contexts. **EP 4** Implementing clustering	

			algorithms using tools like Python, scikit-learn, or similar libraries to process and cluster the data. **EP 5** Designing a system that can classify or profile authors by analyzing Bangla text using unsupervised learning approaches, which requires creativity and advanced application of clustering techniques. **EP 6** Communicating the findings, challenges, and clustering analysis results clearly, whether through reports, presentations, or visualizations. **EP 7** Ensuring that the methods used for author profiling are ethical, respect privacy, and are not biased against any particular author or demographic.	
CO4	Design and develop solutions for the capstone project that meet public health and safety, cultural, societal, and environmental considerations.	**K2:** Understanding the importance of ethical, societal, and cultural factors in designing solutions. **K3:** Applying methods that ensure the project aligns with societal needs and complies with public health and safety standards. **K4:** Reflecting on the social, cultural, and environmental impacts of the project.	EP 1 Demonstrating an understanding of how cultural, societal, and environmental factors affect the development of a technology solution, particularly in the context of Bangla text author profiling. EP 2 Developing solutions that	

		K5: We have designed the use cases for the developed framework. The use case shows the higher-level functionality of the system. We have again depicted the data flow in the diagram (flow diagram).	take into account safety considerations, ensuring the technology doesn't promote misinformation, violate privacy, or harm communities. EP 3 Critically evaluating how clustering methods or author profiling tools might impact public discourse, ethical data use, and the community at large. EP 4 Incorporating appropriate tools or algorithms that are sensitive to cultural nuances in language, ensuring the profiling process respects societal values. EP 5 Designing an ethical, sustainable solution that considers not only technical success but also societal and environmental sustainability, ensuring the system is fair, inclusive, and non-discriminatory. EP 6 Documenting how the project aligns with public health, safety, and cultural considerations, communicating the social responsibility aspects of the project in reports and presentations. EP 7 Ensuring that the data and algorithms used for profiling are ethically	

			sourced, respect user privacy, and do not perpetuate biases or violate societal norms. EP 8 Assessing the broader global impact of the technology solution, including how it may affect diverse populations, and whether it has environmental consequences (e.g., computational resource consumption, energy efficiency).	
CO5	Identify and apply modern engineering and IT tools for the design and development of the capstone project.	K6: We have used all the widely used prevalent python libraries frameworks to validate Machine Learning algorithms. And for developing the developed prototype. We have used Google's Colab for research, Python's flask framework, SK-learn libraries of machine learning, pandas, NumPy. Linux's Ubuntu distribution was used as the operating system. VS code was used in place of a development IDE. GitHub version control for managing the prototype's versions and collaborating with team members.	EP1: For achieving fulfillment, we have researched the modern tools for developing the Framework. We have used Python popular libraries for Machine Learning, Google's researched-based service colab. EP2: Development platform: Google Colab, Language: Python, and Framework: Flask. IDE: Visual Studio Code. EP4: We have used the joblib library for saving our cluster models to reduce time. It is needed when a user wants to find documents quickly. This function helps the initialization time of the model. Furthermore, the TF-IDF's vectors also have been saved in CSV format. EP5: We have used a python flask to prototype with a Machine Learning	

			backend. We have used Colab from google for more GPU power for the development framework. We have used the benchmark dataset Reuters.	
CO6	Assess and address societal, health, safety, legal, and cultural aspects related to implementing the capstone project considering the relevant professional and engineering practices and solutions.	K7: The solution we developed ensures the protection of user data and preferences, allowing users to feel secure when using our application. Additionally, we utilized Google Colab, which supports sustainability and cloud computing for a greener environment. In our text clustering framework, we do not retain any sensitive information. The prototype is designed to avoid political or religious biases, and we have carefully filtered out any sensitive documents.	K7: The solution we developed ensures the protection of user data and preferences, allowing users to feel secure when using our application. Additionally, we utilized Google Colab, which supports sustainability and cloud computing for a greener environment. In our text clustering framework, we do not retain any sensitive information. The prototype is designed to avoid political or religious biases, and we have carefully filtered out any sensitive documents.	

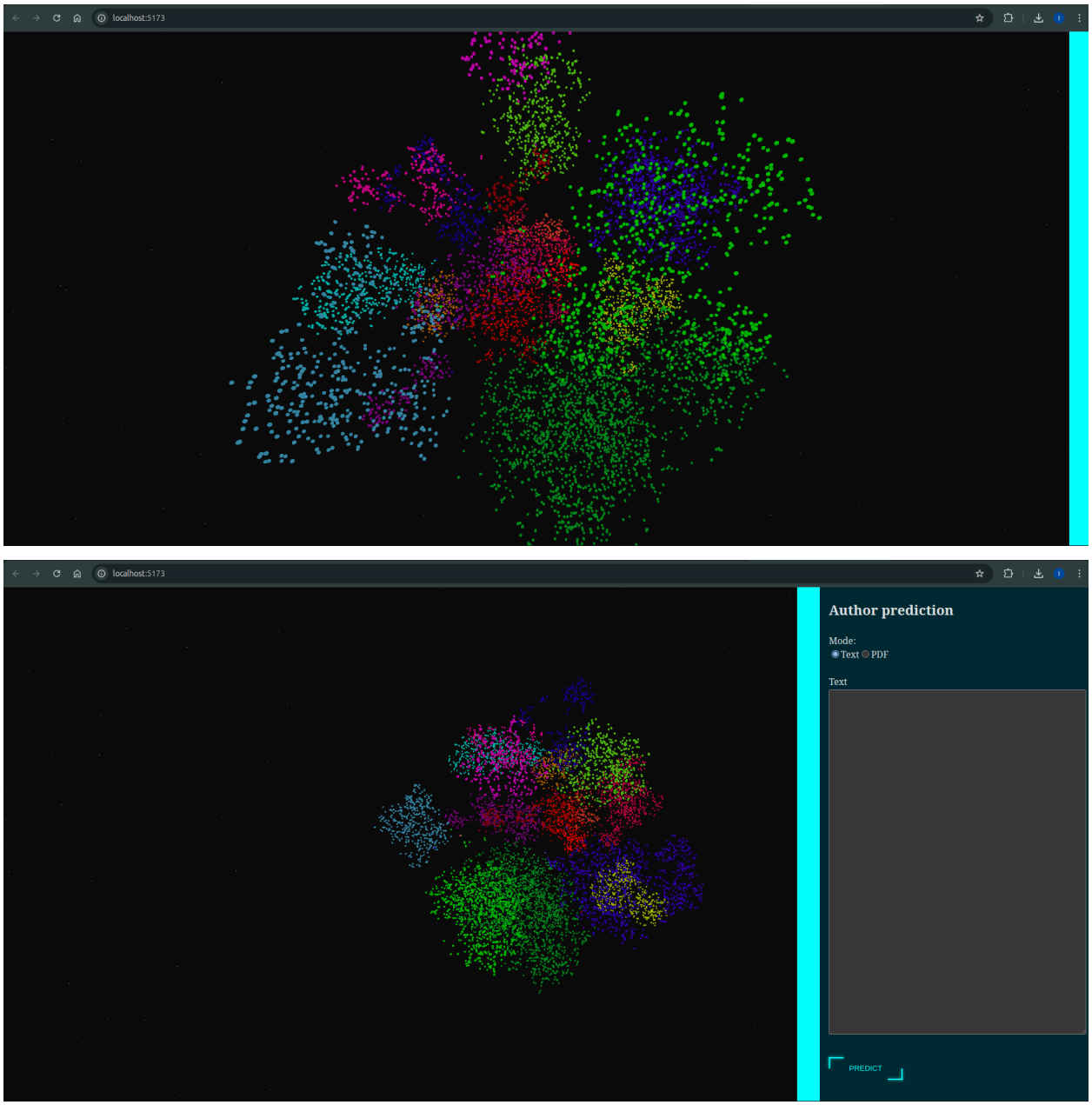
CO	**CO Description**	**K**	**EP/EA**	**PO**
CO7	Our project was developed with environmental sustainability in mind. We have used software-based tools to create a software application, which eliminates the possibility of any negative impact on the environment. Consequently, we can	K7: For Cloud Based Service We used Google Colab		

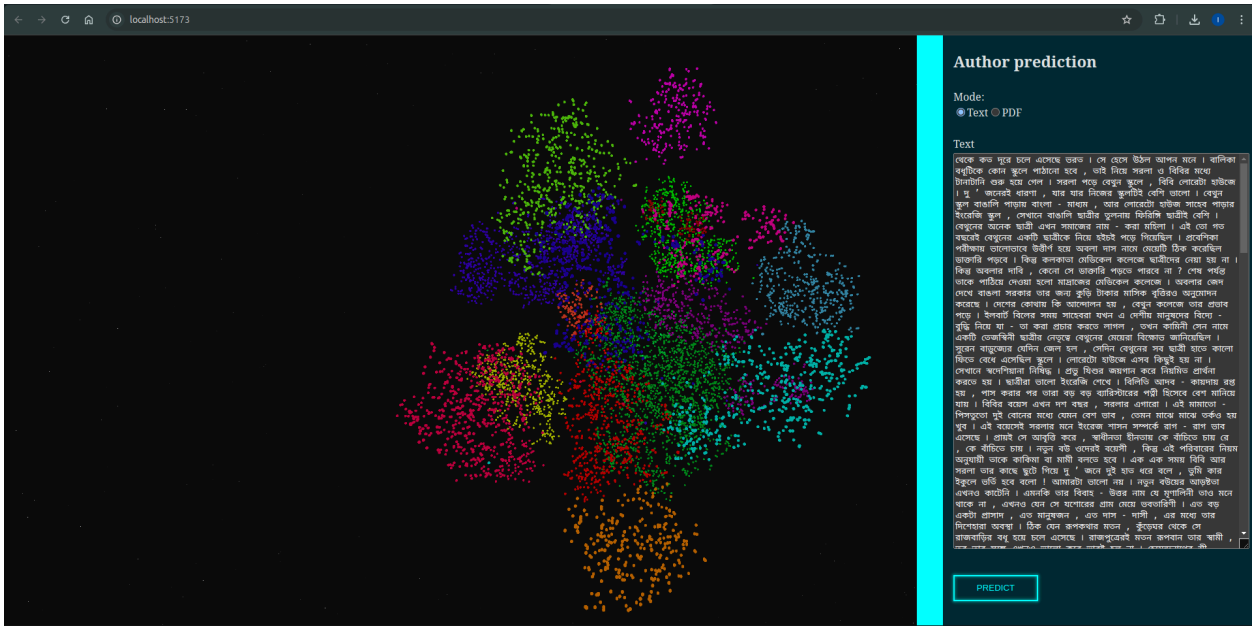
	confirm that our project is eco-friendly.			
CO8	In our project on detecting authors we have paid attention to the ethical implications of our work. As our project requires data from literature of writers of different ethics we have paid a close attention while selecting literatures and avoided any vulgar content. Specifically, we have kept the dataset containing user data completely private and will not share it with any third-party without the user's explicit consent. Furthermore, only the users themselves will have access to view and modify their data within the database. By implementing these measures, we have prioritized the protection of user privacy and ensured that our project meets ethical standards.	We have ensured ethical and public safety issues. We also considered social, cultural, and environment impacts.		
CO9	Work effectively as an individual and a team member for the successful completion of the capstone project		**EP 1**: Demonstrating effective collaboration and communication within a team. **EP 2**: Contributing effectively as an individual to the success of the capstone project. **EP 3**: Balancing individual responsibilities with team objectives to achieve project goals.	

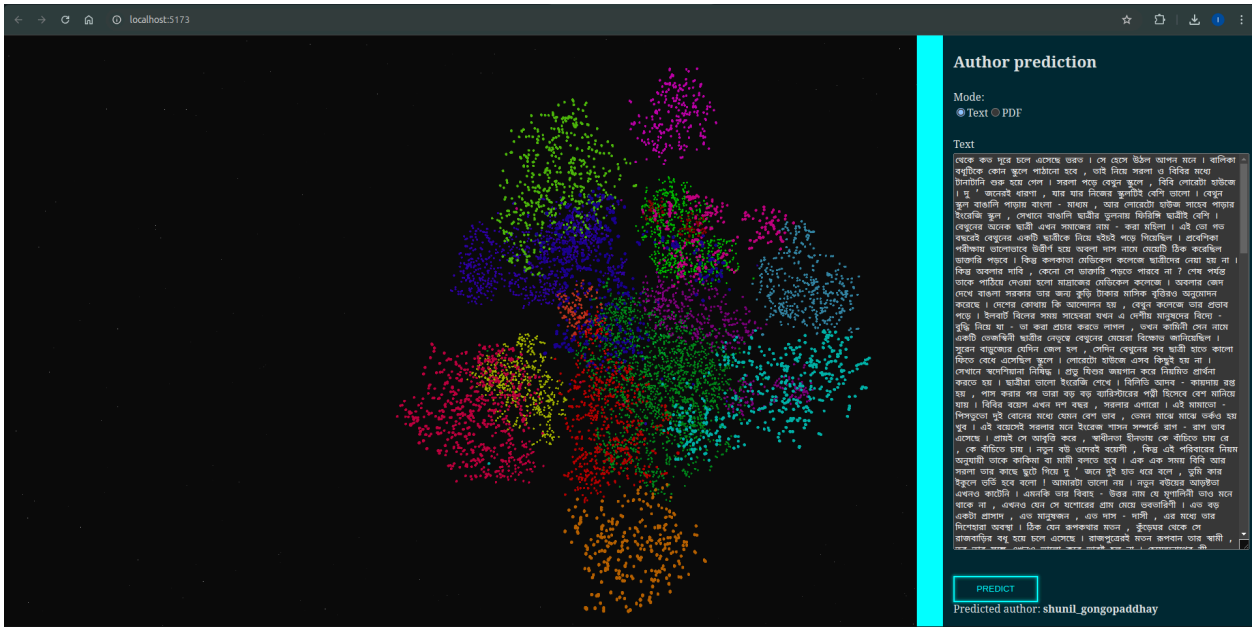
CO10	Write effective reports and design documentation, and make effective presentations of the outcome of the capstone project		**EP 1**: Writing clear and effective reports documenting the project's outcomes and processes. **EP 2**: Designing comprehensive and coherent documentation for project deliverables. **EP 3**: Creating and delivering impactful presentations that communicate the project's results and insights effectively.	
CO11	Conduct economic analysis and cost estimation; and apply appropriate project management processes in the development life cycle of the capstone project			
CO12	Prepare to take part in independent and lifelong learning for adapting emerging technologies for the solution of the complex computer science and engineering problems			

7.2 Appendix B

Stylo-serve frontend:







Google colab notebook:
https://colab.research.google.com/drive/1Icks4hRrtmU-DQdUL2fiYoc0x6IyrkPe?usp=sharing

Frontend Repository: https://github.com/IshmamR/styloviz

Backend Repository:
https://github.com/Azman-Ahmed/Stylometry-based-Author-Profiling-on-Bangla-Text-Using-Text-Clustering