



BANGLA NEWS ARTICLE CLASSIFICATION

PROJECT REPORT

Submitted by

Disha Thakurata	126230100 16
Dishani Ghosh	126230100 17
Miratun Nahar	126230100 20
Saqib Javed	126230100 31
Suvra Bhattacharjee	126230100 54

*in partial fulfillment for 3rd Semester Minor Project
of*

MASTER OF COMPUTER APPLICATION

**DEPARTMENT OF COMPUTER APPLICATIONS
HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA
MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY,
WEST BENGAL
DECEMBER, 2024**



**DEPARTMENT OF COMPUTER APPLICATIONS
HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA**

BONAFIDE CERTIFICATE

Certified that this project report “**Bangla News Article Classification**” is the bonafide work of Disha Thakurata, Dishani Ghosh, Miratun Nahar, Saqib Javed and Suvra Bhattacharjee, students of MCA 3rd Semester of Heritage Institute of Technology, Kolkata, who carried out the project work under the supervision of Prof. Palash Ghosh.

Prof. (Dr.) Souvik Basu
Head,
Department of Computer Applications
Heritage Institute of Technology, Kolkata

Prof. Palash Ghosh
Mentor
Department of Computer Applications
Heritage Institute of Technology, Kolkata

EXAMINER

DECLARATION BY STUDENTS

This is to declare that this report “**Bangla News Article Classification**” has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I confirm that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

Disha Thakurata
Roll No. – 12623010016

Dishani Ghosh
Roll No. – 12623010017

Miratun Nahar
Roll No. – 12623010020

Saqib Javed
Roll No. – 12623010031

Suvra Bhattacharjee
Roll No. – 12623010054

ACKNOWLEDGEMENT

We are deeply grateful to our project guide, Prof. Palash Ghosh, for his constant supervision and valuable guidance throughout the project. We also appreciate Prof. (Dr.) Souvik Basu (HOD, MCA, HITK) for his essential guidance.

A special thanks to Prof. Sandipan Ganguly for his exceptional help and support. We are thankful to the Department of Computer Applications, Heritage Institute of Technology, for providing access to the project lab and facilities.

We also acknowledge Prof. Sumon Ghosh, Prof. Debabrata Kar, and Prof. Anirban Kundu for their valuable support. Lastly, we extend our heartfelt thanks to our mentor, Prof. Palash Ghosh, for his continuous guidance.

Disha Thakurata
Roll No. – 12623010016

Dishani Ghosh
Roll No. – 12623010017

Miratun Nahar
Roll No. – 12623010020

Saqib Javed
Roll No. – 12623010031

Suvra Bhattacharjee
Roll No. – 12623010054

Table of contents

Introduction.....	6
Data Collection.....	7
Data preprocessing.....	9
Features.....	11
Traditional Features.....	11
TERM FREQUENCY (TF).....	11
INVERSE DOCUMENT FREQUENCY (IDF).....	11
TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF).....	11
N-Gram.....	12
Word Cloud.....	12
Deep Learning-Based Features.....	13
Tokenizer.....	13
Text-to-Sequence Conversion.....	13
Padding Sequences.....	14
Classification.....	15
Train-Test Split.....	15
Stratified K-Fold Cross-Validation.....	15
Classifiers.....	16
Traditional Classifiers.....	16
Logistic Regression.....	16
Random Forest.....	17
Naïve Bayes.....	18
Linear Support Vector Machine.....	19
Deep Learning Model.....	20
Long Short Term Memory(LSTM).....	20
Classifiers Comparison.....	22
Metrics used.....	22
Overall Performance.....	23
Class-wise Observations.....	23
Future Scope for Bengali News Classification Project.....	24
Technologies Used.....	25
Conclusion.....	26
References.....	27
Project Repository.....	27

Introduction

The rapid growth of digital media has led to an unprecedented increase in the availability of news articles. Efficiently classifying these articles into relevant categories is crucial for improving information retrieval, content recommendation, and automated content analysis. This project focuses on the development of a Machine Learning (ML)-based system for Bengali news article classification, which represents a significant challenge due to the unique linguistic characteristics and diverse styles of the Bengali language.

Bengali, being the seventh most spoken language in the world, exhibits a rich and complex linguistic structure with intricate grammar, extensive use of compound words, and stylistic variations across different forms of writing. Despite the global advancements in text classification systems for languages such as English, Spanish, and Chinese, Bengali remains underexplored in this domain.

In this project, we address the task of categorizing Bengali news articles into predefined categories such as National, Science, Education, International, Sports, Politics, Kolkata, and State. This classification problem is particularly challenging due to the significant class imbalance in the dataset, as some categories (e.g., National and Science) have a large number of articles, while others (e.g., State and Kolkata) have relatively fewer samples.

To tackle these challenges, we utilize various Machine Learning models to analyze the linguistic patterns and semantic features of Bengali news articles. Our goal is to build an efficient and scalable classification system that demonstrates high performance in terms of accuracy, precision, recall, and F1-score across all categories. By addressing this problem, we aim to contribute to the development of NLP tools for low-resource languages like Bengali, thereby enhancing the accessibility and usability of digital information in this language.

Data Collection

For this project, we collected Bengali news articles from three publicly available datasets to create a comprehensive and diverse corpus:

i. Shironaam Dataset ([Hugging Face](#)):

This dataset contains a collection of Bengali news headlines, providing concise and informative text samples from various categories.

Shironaam :						
	news_link	head_lines	article	tags	image_caption	category
128230	https://samakal.com/whole-country/article/2004...	বগুড়ায় ৩৩০ কোজি চালসহ বিএনপিপন্থী ইউপি চেয়ারম্...	কমহীনদের জন্য সরকারের বরাদ্দের চাল চুরির অভি...	বগুড়া,শিবগঞ্জ,চেয়ারম্যান গ্রেফতার	মির্জা গোলাম হাফিজ সোহাগ	national
206281	https://www.bhorerkagoj.com/2020/09/15/%e0%a6%...	রিকশা, ভ্যান ও ঠেলাগাড়ী বন্ধ না করার দাবি	বাংলাদেশের ওয়ার্কার্স পার্টি ঢাকা মহনগর কমিটির...	NaN	ওয়ার্কার্স পার্টি	national
180523	https://www.dailynayadiganta.com/mymensingh/35...	শেরপুরে ফুলছাত্তীকে গণধর্ষণের পর হত্যা : ৩ জ...	শেরপুরের বিনাইগাতী উপজেলার বাকানুড়া গ্রামে ৫ম ...	ধর্ষণ,হত্যা,মৃত্যুদণ্ড	রায় ঘোষণার পর আমানুল্লা ও নুরে আলমকে কারাগারে ...	national

ii. Bengali News Articles Dataset ([Kaggle](#)):

This dataset offers a rich collection of Bengali news articles sourced from renowned newspapers like **Anandabazar Patrika**, **Ebela**, and **Zee News**, categorized into multiple predefined classes. It covers various domains such as politics, sports, and international news, offering linguistic diversity and stylistic richness.

Anandabazar, Ebela, Zee News :			
	title	article	label
0	এই অভিনেতার 'প্রভাব' মানলে ভারতীয় সিনেমার খো...	শ্রেফ দু'টি টুইটেই সোশ্যাল মিডিয়ায় তোলপাড় ফে...	entertainment
1	সচিন সাংসদ হয়েছেন, তাই চিন্তা কম জেটলির। কেন?	একবার রেল বাজেট। একবার সাধারণ বাজেট। খবরের শি...	national
2	ক্রিকেটার থেকে গাইড: জাতীয় দলের ক্রিকেটারের অব...	কিছুদিন আগেই রাজ্যের গাইড বনে গিয়েছিলেন চেতেশ...	sports
3	দুর্কৃতীদের নজরে এটিএম, হাল হকিকত ঘুরে দেখলেন...	কলকাতায় দুর্কৃতীদের নজরে এখন বিভিন্ন ব্যাঙ্কের...	kolkata
4	সৌজন্যের বালাই নেই দিনভর আকচা আকচি কং-বিজেপি র	সারা দিন আকচা আকচিতেই কাটিয়ে দিল দেশের দুই বৃহ...	national

iii. Potrika Bangla Newspaper Dataset ([Kaggle](#)):

This dataset includes Bengali news articles with a focus on in-depth reporting and coverage of diverse topics, adding linguistic variety to our corpus.

Potrika :		
	article	class
229523	জাতীয় পার্টির কোচেরারম্যার জিএম কাদের জাতীয় পা...	Politics
207477	আওয়ামী লীগ আসন্ন ইউনিয়ন পরিষদ নির্বাচনে চেয়...	Politics
296452	আইপিএলদিল্লিমুন্সাই বিকাল টাচেমাইপাঞ্জাব রাত...	Sports
213850	ক্ষমতাসীন আওয়ামী লীগ সংসদ নির্বাচনকে একতরফা ক...	Politics
68725	সহজ প্রশ্নসময় ঘণ্টা মিনিট পূর্ণ দৃষ্টব্য প্র...	Education

After collecting the data from these sources, we performed several preprocessing steps to ensure consistency, quality, and usability of the dataset:

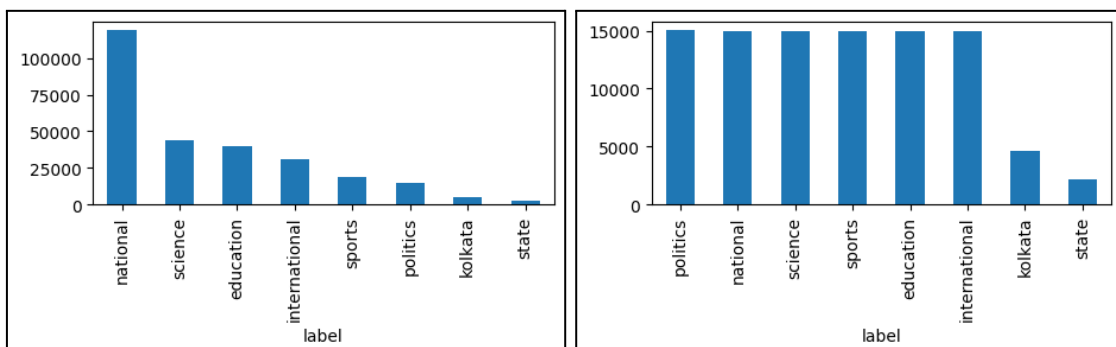
- **Data Cleaning:** Removed duplicates, irrelevant entries, and inconsistencies such as missing labels or corrupted text.

Anandabazar, Ebela, Zee News :			Potrika :		
	article	label		article	label
1643	তৃণমূলনেতা মুকুল রায়ের পর এবার তাঁর ছেলে শুভ...	state	252140	মধ্যপ্রাচ্য ইতালি মার্কিন স্বাধীনতাসংগ্রাম জলদ...	science
9620	প্রথমে দেখা গিয়েছিল একজনকে। তাও সে আকারে নেহা...	international	71929	চতুর্থ প্রজন্মের প্রযুক্তিই ফোরজি তারবিহীন প্র...	education
5122	বিয়ের আগের সম্পর্ক। বিবাহ-বহির্ভূত সম্পর্ক। ল...	national	62491	মাসের হার সুদে টাকা উত্তোলন বার্ষিক উত্তোলনের ...	education
Shironaam :					
	article	label		article	label
9264	ঢাকা বিশ্ববিদ্যালয়ের (ঢাবি) শতবর্ষ এবং ইতিহাস ...	national			
97917	পাকিস্তানে সেনাপ্রধানকে বরখাস্ত করার বিষয়ে বিব...	international			
208529	জমিসহ আধা-পাকা দোকানঘর দখল করতে বরিশালের উজিরপ...	national			

- **Merging:** Combined the cleaned data from all three datasets into a single unified dataset, encompassing a wide range of linguistic styles and topics.

Merged Dataset		
	article	label
88668	দৈনন্দিন জীবনযাত্রাকে একদমই পাল্টে দেবে ফাইভজি...	science
205980	ঢাকা দক্ষিণ সিটি করপোরেশন নির্বাচনে মেয়র পদে ব...	politics
260165	চীনের প্রতিষ্ঠা বার্ষিকীর দিনে একদল উইঘুর সম্প...	international
17758	ভৌগোলিক কারণে বাংলাদেশের বেশির ভাগ অঞ্চলক বেল...	education
177526	লেখক ও অধ্যাপক ড. মুহম্মদ জাফর ইকবাল বলেছেন, ই...	national

- **Balancing:** Addressed the issue of class imbalance by carefully curating the samples for under-represented categories to ensure that the model does not favor over-represented classes.



The final dataset provides a robust foundation for training and evaluating machine learning models for Bengali news article classification. It ensures a diverse representation of topics while maintaining data quality and balance across categories.

Data preprocessing

To prepare the collected data for machine learning model training, we performed several preprocessing steps to clean and standardize the text. The key preprocessing techniques used in this project are:

i. **Text Cleaning Functions:**

Several custom functions were used to remove specific unwanted elements from the text:

- **Remove English Words:**

A function was created to remove English words embedded within Bengali text using the following regex pattern:

```
def removeEnglish(text):  
    cleaned_text = re.sub(r'\b[A-Za-z]+(?:\s+)*\b(?:\s+)*[A-Za-z]+\b', '', text)  
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()  
    return cleaned_text
```

- **Remove brackets:**

A function was created to remove brackets, such as "()", {}, that do not contribute to the meaning of the article:

```
def remove_brackets(text):  
    return re.sub(r'\(.*?\)|\{.*?\}', '', text)
```

- **Remove Bengali Numerals:**

Another function was applied to remove Bengali numerals (U+09E6 to U+09EF Unicode block), as they may not be relevant for the classification task:

```
def remove_bengali_numerics(text):  
    return re.sub(r'[\u09E6-\u09EF]', '', text)
```

ii. Category Encoding using Label Encoder:

Since the dataset contains categorical labels for news categories (e.g., National, Science, Sports), we used a **Label Encoder** to convert these text labels into numerical values. This transformation ensures that the machine learning models can efficiently handle categorical variables during the training process.

	article	label	category_encoded
30668	মিরপুর টেস্টের তৃতীয় দিনের শুরুতেই অলআউট হয়ে য...	sports	6
93339	প্রস্তুত ছিল না বহরমপুর। কতটা প্রস্তুত কলকাতা...	kolkata	2
36904	লা লিগায় ইতিহাস গড়লেন আর্জেন্টাইন ফুটবল জাদুকর...	sports	6
61814	দেশদ্রোহে অভিযুক্ত প্রাক্তন প্রেসিডেন্টের খামা...	international	1
45552	উদ্দীপনা পড়ুনরাকিব সারাদিন টিভি দেখার নষ্ট জী...	education	0

iii. Bengali Stop Words Removal:

To reduce the noise in the text data, we removed common Bengali stopwords—words that carry little meaning and are typically removed in natural language processing (NLP) tasks. These words, such as "এটা", "আর", "ও", do not contribute significant value to the text's meaning and can distort the model's ability to learn meaningful patterns. Removing these stopwords helps to focus the model's attention on more informative and content-rich words, which ultimately enhances the classification accuracy.

For the stop words removal process, we utilized two primary resources:

- Bengali Stop Words List:** We sourced a comprehensive list of Bengali stopwords from a publicly available repository on **GitHub**. This list contains hundreds of frequently occurring, low-information words that are typically discarded in text preprocessing.
- Custom Excel (XLSX) File:** In addition to the GitHub list, we also used a custom **XLSX file** that includes an extended list of Bengali stopwords specifically curated for our dataset. This file was carefully selected to cover domain-specific terms that could be considered stopwords in the context of Bengali news articles.

Features

Traditional Features

TERM FREQUENCY (TF)

Term Frequency (TF) measures the frequency of a specific word in each news article within the dataset. It is calculated as the ratio of the number of times a word appears in a given article compared to the total number of words in that article. A higher TF indicates that the word occurs more frequently within that particular article. This feature helps capture the relevance of specific words within individual news articles.

The TF can be calculated as:

$$TF = \frac{\text{Number of times word appears in the article}}{\text{Total number of words in the article}}$$

For example, in a **Science** news article, terms like "বিজ্ঞান" (science) or "প্রযুক্তি" (technology) might have high TF scores if they are repeated multiple times in the context of the article.

INVERSE DOCUMENT FREQUENCY (IDF)

Inverse Document Frequency (IDF) measures how common or rare a word is across the entire collection of Bengali news articles. It is calculated by taking the logarithm of the total number of articles divided by the number of articles that contain the word. If a word appears in many articles, its IDF score will be close to 0, indicating it is common. Conversely, if a word appears in only a few articles, its IDF score will be higher, indicating it is rare and potentially more important for distinguishing between different categories of news.

The IDF for a given word is calculated as:

$$IDF = \log\left(\frac{\text{Total number of articles}}{\text{Number of articles containing the word}}\right)$$

For example, in a **Politics** news article, "রাজনীতি" (politics) might have a higher IDF score, indicating it is a unique and significant term in the context of political articles, but less likely to appear in articles about **Sports** or **Science**.

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

TF-IDF is a combined metric that multiplies the scores of Term Frequency (TF) and Inverse Document Frequency (IDF). This score represents the importance of a word within a specific news article, taking into account both how often it appears within the article and how unique or rare it is across all articles in the dataset. High TF-IDF scores indicate that a word is not only frequent in a particular article but also contributes significantly to distinguishing that article from others. TF-IDF is used extensively in our classification models, as it improves the performance of algorithms like Naïve Bayes and Support Vector Machines, especially when compared to simpler word count-based methods.

The TF-IDF score is calculated as:

$$TF\text{-}IDF = TF * IDF$$

Deep Learning-Based Features

Tokenizer

The Tokenizer from *tensorflow.keras* is used to preprocess Bengali news articles into a format suitable for deep learning models. It handles:

Vocabulary Creation

The Tokenizer is initialized with a maximum vocabulary size suitable for the context, (e.g. `num_words=41000`) to ensure it includes the most frequent words across all articles.

The `tk.fit_on_texts(data['article'])` method processes the text data in the dataset (`data['article']`) and:

- Builds a vocabulary of unique words, assigning each a unique integer index based on its frequency (more frequent words receive lower indices).
- Ignores words outside the top 41,000 most frequent ones to focus on important terms and reduce computational overhead.

```
tk = Tokenizer(num_words=41000)
tk.fit_on_texts(data['article'])
```

```
print("First 10 Words in Vocabulary:", list(tk.word_index.items())[100:150])
```

First 10 Words in Vocabulary: [('কাঁচা', 101), ('মেতা', 102), ('নির্বাচন', 103), ('মার্কিন', 104), ('ম্যাচ', 105), ('প্রসিডেন্ট', 106), ('উপজেলা', 107), ('ব লেখিলেন', 108), ('হাসান', 109), ('বিশ্ববিদ্যালয়ের', 110), ('কর্মকর্তা', 111), ('ছবি', 112), ('দেশ', 113), ('হাতে', 114), ('একটা', 115), ('অব', 116), ('ঘটনা', 117), ('বিশ্বায়', 118), ('পড়ে', 119), ('আগের', 120), ('লীগ', 121), ('আলোচনা', 122), ('সরকারি', 123), ('সম্পর্কে', 124), ('সবচেয়ে', 125), ('সোমবার', 126), ('মঙ্গলবার', 127), ('বৃহস্পতিবার', 128), ('দিনের', 129), ('মাঠে', 130), ('অভিনয়', 131), ('বাংলাদেশে', 132), ('টাকার', 133), ('বুধবার', 134), ('এম', 135), ('এরপর', 136), ('এসিয়ে', 137), ('প্রযুক্তি', 138), ('রাজধানীর', 139), ('বাজারে', 140), ('শনিবার', 141), ('কোনাট', 142), ('আলম', 143), ('প্রতিষ্ঠান', 144), ('রাজনৈতিক', 145), ('পয়েন্ট', 146), ('সকাল', 147), ('আলী', 148), ('সময়ে', 149), ('স্থানীয়', 150)]

Text-to-Sequence Conversion

After the vocabulary is created, the text data is converted into numerical sequences using `tk.texts_to_sequences(data['article'])`.

- Each article is represented as a sequence of integer indices corresponding to the words in the text.
- Words not in the tokenizer's vocabulary are ignored or replaced with a special token.
- Encodes text data into a format that captures the frequency and order of words, making it suitable for input into the deep learning model.

```
seq = tk.texts_to_sequences(data['article'])
```

```
print(data['article'][10])
print("Sequences:", seq[10])
```

ফেব্রুয়ারি তুলনার মার্চ মাসে সার্বিক মূল্যস্ফীতি বেড়েছে মার্চ মাসে মূল্যস্ফীতি বৃদ্ধির হার পয়েন্ট টু পয়েন্ট ঊর্দ্ধে দাঁড়িয়েছে দশমিক শতাংশে ফেব্রুয়ারিতে হার দশমিক শতাংশ খাদ্যপণ্যের দাম বাড়তে সার্বিক মূল্যস্ফীতিতে প্রভাব পড়েছে কমেছে খাদ্যবাহিত পণ্যের মূল্যস্ফীতিমঙ্গলবার রাজধানীর শেরেবাংলা নগরের এনইসি সম্মেলন কক্ষে পরিসংখ্যান ব্যুরোর প্রকাশ প্রতিবেদন তথ্য জানান পরিকল্পনামন্ত্রী আ হ ম মুস্তফা কামাল প্রকাশিত প্রতিবেদন পর্যালোচনার যায় খাদ্যপণ্যের মূল্যস্ফীতি বেড়ে দাঁড়িয়েছে দশমিক শতাংশে আগের মাসে হার দশমিক শতাংশ কমেছে খাদ্যবাহিত পণ্যের মূল্যস্ফীতি আগের মাসের দশমিক শতাংশ কমে হার দশমিক শতাংশে নেমে এসেছেব্রিফিংয়ে প্রসঙ্গে চাইলে পরিকল্পনামন্ত্রী উদীয়মান অর্থনীতির দেশে মূল্যস্ফীতি একেবারেই কমানো না তাছাড়া আগের মাসের তুলনার মার্চ মাসের তুলনার মার্চ জিনিসপত্রের দাম বেড়েছে ইলিশ মাছের দাম বেড়েছেএছাড়া গরুর মাংস চিনির দামও বেড়েছে এসব কারণেই মূল্যস্ফীতি বৃদ্ধি পেয়েছেবিএসএর প্রতিবেদন গ্রামে সার্বিক মূল্যস্ফীতি পয়েন্ট টু পয়েন্ট ভিত্তিতে বেড়ে দাঁড়িয়েছে দশমিক শতাংশে আগের মাসে দশমিক শতাংশ খাদ্যপণ্যের মূল্যস্ফীতি বেড়ে দাঁড়িয়েছে দশমিক শতাংশে আগের মাসে দশমিক শতাংশ খাদ্যবাহিত পণ্যের মূল্যস্ফীতি কমে দাঁড়িয়েছে দশমিক শতাংশে আগের মাসে দশমিক শতাংশ খাদ্যপণ্যের মূল্যস্ফীতি বেড়ে দাঁড়িয়েছে দশমিক শতাংশে আগের মাসে দশমিক শতাংশ খাদ্যবাহিত পণ্যের মূল্যস্ফীতি কমে দাঁড়িয়েছে দশমিক শতাংশে আগের মাসে দশমিক শতাংশ

Sequences: [9886, 575, 281, 166, 1176, 4961, 260, 281, 166, 4961, 1105, 341, 146, 1390, 146, 556, 814, 198, 3163, 18020, 341, 198, 34, 10418, 78, 2460, 1176, 513, 479, 486, 18021, 619, 139, 3500, 6246, 8842, 468, 1326, 2517, 6291, 61, 571, 27, 7, 5674, 887, 4082, 580, 4264, 350, 476, 571, 7943, 10, 104, 18, 4961, 342, 814, 198, 3163, 120, 166, 341, 198, 34, 486, 18021, 619, 4961, 120, 298, 198, 34, 374, 341, 198, 3163, 407, 420, 393, 5674, 7069, 1735, 41, 4961, 2984, 3464, 1, 2245, 120, 298, 575, 3780, 18758, 78, 260, 3838, 3486, 78, 3599, 4169, 6882, 5758, 260, 37, 847, 4961, 304, 571, 785, 1176, 49, 61, 146, 1390, 146, 556, 342, 814, 198, 3163, 120, 166, 198, 34, 10418, 4961, 342, 814, 198, 3163, 120, 166, 198, 15718, 34, 18021, 619, 4961, 374, 81, 4, 198, 3163, 120, 166, 198, 34, 441, 901, 1176, 4961, 146, 1390, 146, 556, 342, 814, 198, 3163, 120, 166, 198, 34, 10418, 4961, 342, 814, 198, 3163, 120, 166, 198, 34, 18021, 619, 4961, 374, 814, 198, 3163, 120, 166, 198, 34]

Padding Sequences

The sequences are standardized to a fixed length (e.g. of 200) using **pad_sequences(seq, padding='post', maxlen=200)**. This ensures all inputs to the model are of uniform size:

- Handles variable-length news articles.
- Maintains sentence structure by padding/truncating from the end.

```
vec = pad_sequences(seq, padding='post', maxlen=200)
```

```
print("Padded Sequences:")
print(vec)
```

```
Padded Sequences:
[[23252 285 164 ... 472 1567 409]
 [ 3762 5839 5825 ... 0 0 0]
 [13938 107 1889 ... 0 0 0]
 ...
 [ 6103 585 4158 ... 0 0 0]
 [ 71 7223 1689 ... 1555 37441 10484]
 [ 163 1394 95 ... 0 0 0]]
```

The final padded sequences (vec) serve as input to the LSTM model. These representations capture:

- Word Frequency and Importance: Through the tokenizer's numerical mapping of frequently occurring words.
- Sequential Context: By preserving the order of words in each sequence, enabling the LSTM to learn dependencies over time.

Classification

Train-Test Split

After preprocessing the dataset, we performed a train-test-validation split to ensure the data was properly partitioned for training and evaluation. Specifically, we allocated:

- **50% of the data for training:** This portion was used to train the machine learning models.
- **25% for testing:** This set was used to evaluate the performance of the trained models and determine their generalization to unseen data.
- **25% for validation:** This set helped fine-tune model parameters and prevent overfitting.

```
X_train, X_temp, y_train, y_temp = train_test_split(
    balanced_df['article'], balanced_df['category_encoded'], test_size=0.5,
    random_state=35, stratify=balanced_df['category_encoded']
)
```

50% Train Dataset

```
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, random_state=35, stratify=y_temp
)
```

25% Validation, 25% Test Datasets

The train-test split ensured that the models were trained and evaluated on separate portions of the data, providing a reliable measure of performance for Bengali news classification.

Stratified K-Fold Cross-Validation

To further validate the models and address potential bias or variance, we used **Stratified K-Fold Cross-Validation**. This technique extends the standard k-fold cross-validation by ensuring that each fold preserves the original class distribution of the dataset.

In the context of Bengali news classification, where categories like **National** and **Sports** might have different sample sizes, stratification ensures that each fold contains a representative mix of all categories. This helps the models learn effectively from each category and provides more robust performance metrics.

For example:

If a particular category like **State News** has fewer samples compared to **National News**, the stratified approach ensures that the minority class is proportionately represented in all folds. This prevents the model from being biased toward overrepresented categories and improves its ability to generalize across all news categories.

By combining the train-test split with stratified k-fold cross-validation, we ensured a balanced and rigorous evaluation of our machine learning models for Bengali news classification.

```
kf = StratifiedKFold(n_splits=5, shuffle=True, random_state=35)
```

Classifiers

Traditional Classifiers

Logistic Regression

Logistic Regression is a linear model that predicts the probability of a news article belonging to a specific category. It is efficient for text classification tasks like Bengali news classification, where it maps input features (e.g., TF-IDF vectors) to categorical labels.

```
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier

logistic_model = OneVsRestClassifier(LogisticRegression(max_iter=1000,
    solver='lbfgs'))
```

The obtained result is as follows:

	precision	recall	f1-score	support
education	0.93	0.94	0.94	3750
international	0.87	0.91	0.89	3750
kolkata	0.82	0.84	0.83	1157
national	0.81	0.76	0.79	3750
politics	0.84	0.85	0.85	3754
science	0.93	0.93	0.93	3750
sports	0.95	0.97	0.96	7500
state	0.82	0.49	0.62	550
accuracy			0.90	27961
macro avg	0.87	0.84	0.85	27961
weighted avg	0.89	0.90	0.89	27961

The logistic regression model achieved an overall **accuracy of 90%** for Bengali news classification across eight categories. Key observations include:

- **High Precision and Recall:** Categories like **Education (F1-score: 0.94)**, **Science (F1-score: 0.93)**, and **Sports (F1-score: 0.96)** demonstrated excellent performance, indicating the model's effectiveness in correctly classifying these well-represented categories.
- **Moderate Performance:** Categories like **Kolkata (F1-score: 0.83)** and **Politics (F1-score: 0.85)** showed moderate performance, reflecting the challenges posed by overlapping content with other categories.
- **Low Performance for Minor Categories:** The **State category (F1-score: 0.62)**, with fewer samples, had lower precision and recall, highlighting the model's struggle with imbalanced data.

Random Forest

Random Forest is an ensemble learning method that uses multiple decision trees to classify Bengali news articles. It enhances accuracy and reduces overfitting by combining predictions from various trees, making it robust for large and diverse datasets.

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=35)
```

The obtained result is as follows:

	precision	recall	f1-score	support
education	0.91	0.92	0.92	3750
international	0.77	0.88	0.82	3750
kolkata	0.84	0.49	0.62	1157
national	0.82	0.67	0.74	3750
politics	0.81	0.86	0.83	3754
science	0.91	0.91	0.91	3750
sports	0.88	0.98	0.93	7500
state	0.98	0.08	0.15	550
accuracy			0.85	27961
macro avg	0.87	0.72	0.74	27961
weighted avg	0.86	0.85	0.84	27961

The Random Forest model achieved an overall **accuracy of 85%** for Bengali news classification across eight categories. Key observations include:

- **Strong Performance:** Categories such as **Education (F1-score: 0.92)**, **Science (F1-score: 0.91)**, and **Sports (F1-score: 0.93)** were classified effectively, showcasing the model's ability to handle well-represented and distinct categories.
- **Moderate Challenges:** Categories like **International (F1-score: 0.82)** and **Politics (F1-score: 0.83)** exhibited moderate performance, reflecting the model's capability to handle slightly overlapping content.
- **Struggles with Minority Classes:** The **State category (F1-score: 0.15)** performed poorly due to its small dataset size, indicating limitations in handling under-represented categories. Similarly, **Kolkata (F1-score: 0.62)** showed lower recall, suggesting difficulties in distinguishing it from other categories.
- **Imbalance Effect:** The macro average metrics (**Precision: 0.87**, **Recall: 0.72**, **F1-score: 0.74**) highlight the model's reduced recall for minority classes compared to weighted averages.

Naïve Bayes

Naïve Bayes is a probabilistic classifier based on Bayes' Theorem, assuming independence between features. It is computationally efficient and well-suited for text classification tasks like Bengali news classification, particularly with categorical data like word counts or TF-IDF scores.

```
from sklearn.naive_bayes import MultinomialNB

nb_model = MultinomialNB()
```

The obtained result is as follows:

	precision	recall	f1-score	support
education	0.93	0.87	0.90	3750
international	0.80	0.90	0.85	3750
kolkata	0.71	0.68	0.69	1157
national	0.75	0.71	0.73	3750
politics	0.77	0.85	0.81	3754
science	0.93	0.90	0.92	3750
sports	0.92	0.96	0.94	7500
state	1.00	0.01	0.01	550
accuracy			0.85	27961
macro avg	0.85	0.74	0.73	27961
weighted avg	0.86	0.85	0.85	27961

The Naïve Bayes classifier achieved an overall accuracy of **85%** on the test dataset. It performed well in categories like **Sports** (F1-score: 0.94), **Science** (F1-score: 0.92), and **Education** (F1-score: 0.90). However, it struggled significantly with the **State** category, achieving only a **1% F1-score**, indicating difficulty in handling underrepresented classes or imbalanced data.

The model's relatively strong performance in other categories demonstrates its suitability for text classification tasks where word-based probabilistic relationships play a key role. Nonetheless, it highlights the need for advanced methods or preprocessing techniques to address imbalanced datasets for improved classification across all categories.

Linear Support Vector Machine

SVM is a supervised learning algorithm that classifies Bengali news articles by finding the optimal hyperplane that separates different categories. It works well with high-dimensional data like TF-IDF vectors and is effective for both linear and non-linear classification tasks.

```
from sklearn.svm import SVC

svm_model = SVC(kernel='linear', C=1.0) # Linear Kernel for text classification
```

The obtained result is as follows:

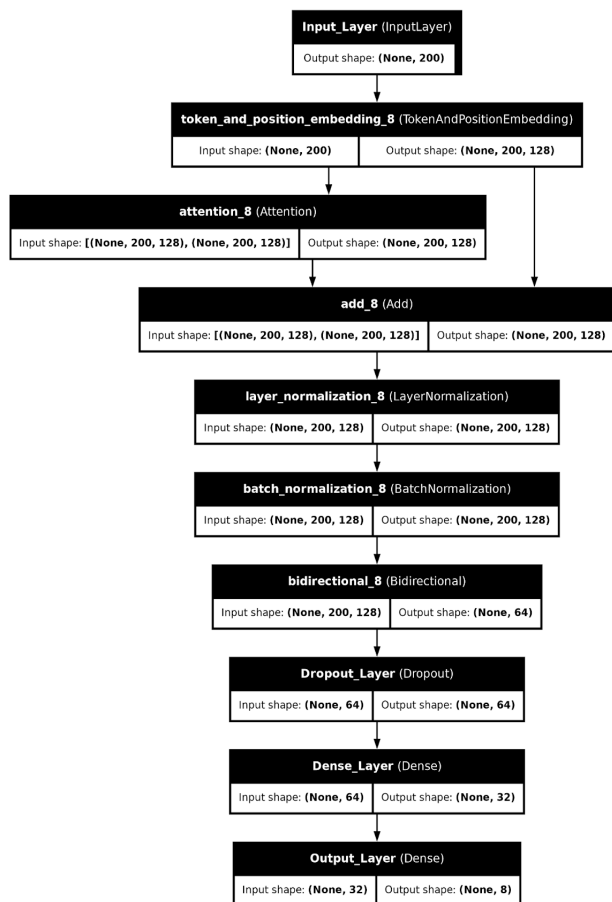
	precision	recall	f1-score	support
education	0.95	0.94	0.94	3750
international	0.88	0.94	0.91	3750
kolkata	0.84	0.87	0.85	1157
national	0.82	0.76	0.79	3750
politics	0.83	0.86	0.85	3754
science	0.94	0.94	0.94	3750
sports	0.96	0.95	0.96	3750
state	0.77	0.66	0.71	550
accuracy			0.89	24211
macro avg	0.87	0.86	0.87	24211
weighted avg	0.89	0.89	0.89	24211

The Support Vector Machine (SVM) model achieved an overall **accuracy of 89%** for Bengali news classification across eight categories. Key observations include:

- **High Performance:** Categories like **Education (F1-score: 0.94)**, **Science (F1-score: 0.94)**, and **Sports (F1-score: 0.96)** demonstrated excellent precision and recall, reflecting the SVM's capability to separate distinct categories effectively.
- **Moderate Performance:** Categories such as **International (F1-score: 0.91)**, **Politics (F1-score: 0.85)**, and **Kolkata (F1-score: 0.85)** performed well, though with slight overlap in content affecting recall.
- **Challenges with Minority Classes:** The **State category (F1-score: 0.71)** had lower recall (66%), indicating difficulty in distinguishing under-represented categories. The overall macro average metrics (**F1-score: 0.87**) highlight this disparity.
- **Consistent Weighted Performance:** The weighted averages (**F1-score: 0.89**) show the SVM's balanced performance across categories, particularly for more frequent labels like **Sports** and **National**.

Deep Learning Model

Long Short Term Memory(LSTM)



This model is designed for sequence processing tasks, leveraging a combination of attention mechanisms and bidirectional recurrent layers. Below is an overview of the main components:

1. **Input Layer:** Accepts input sequences, padded sequences (vec) with a length of , for example ,200. The shape is `(None, 200)`, where `None` allows for variable batch sizes.
2. **Token and Position Embedding Layer:** This layer embeds the input tokens into dense vectors and includes positional encoding to capture the order of the tokens in the sequence. The output shape is `(None, 200, 128)`.
3. **Attention Layer:** A multi-head attention mechanism is applied to capture the dependencies between tokens in the input sequence. It operates on the shape `(None, 200, 128)` and outputs a tensor of the same shape.
4. **Add Layer:** This layer adds the attention output to the input embeddings, allowing the model to learn residual connections. The output shape remains `(None, 200, 128)`.
5. **Normalization Layers:**
 - **Layer Normalization:** This normalizes the input across the feature axis to improve training stability.
 - **Batch Normalization:** Normalizes the activations during training to reduce internal covariate shift, improving generalization.
6. **Bidirectional LSTM Layer:** A bidirectional LSTM layer processes the sequence in both forward and reverse directions, capturing contextual information from both past and future tokens. The output shape is `(None, 64)`.
7. **Dropout Layer:** A dropout layer is applied with a rate of 0.7 to reduce overfitting by randomly setting a fraction of input units to zero during training.
8. **Dense Layer:** A fully connected dense layer with 32 units, which processes the features extracted by the recurrent layers.
9. **Output Layer:** The final layer is a dense layer with 8 output units, corresponding to the number of classes for classification tasks. The output shape is `(None, 8)`.

The obtained result is as follows:

	.precision	recall	f1-score	support
0	0.88	0.87	0.88	4091
1	0.92	0.93	0.93	3932
2	0.95	0.97	0.96	4077
3	0.92	0.94	0.93	4046
4	0.89	0.94	0.91	4041
5	0.92	0.93	0.93	4255
6	0.98	0.98	0.98	4137
7	0.84	0.76	0.80	4039
accuracy			0.91	32618
macro avg	0.91	0.91	0.91	32618
weighted avg	0.91	0.91	0.91	32618

The LSTM (Long Short-Term Memory) model achieved an overall accuracy of **91%** in classifying Bengali news articles. It performed exceptionally well in categories like **Sports** (F1-score: 0.98), **National** (F1-score: 0.96), and **Education** (F1-score: 0.93). However, it faced challenges in the **State** category, where the F1-score was comparatively lower at **0.80**.

The model effectively captured long-term dependencies in the text, demonstrating its ability to handle the sequential nature of Bengali language articles. Its robust performance across most categories highlights the suitability of deep learning approaches like LSTM for complex classification tasks in low-resource languages.

Classifiers Comparison

This project compares five machine learning models—**SVM, Naive Bayes, Random Forest, Logistic Regression, and LSTM**—for classifying Bengali news articles into eight categories: Education, International, Kolkata, National, Politics, Science, Sports, and State.

Metrics used

Accuracy

- **Definition:** The ratio of correctly predicted samples to the total number of samples.
- **Formula:**

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

- **Purpose:** Provides an overall performance summary but can be misleading for imbalanced datasets.

Precision

- **Definition:** The ratio of correctly predicted positive observations to the total predicted positive observations.
- **Formula:**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Purpose:** Indicates the model's ability to avoid false positives, critical in applications where incorrect classification has high consequences.

Recall (Sensitivity)

- **Definition:** The ratio of correctly predicted positive observations to all actual positive observations.
- **Formula:**

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Purpose:** Measures the model's ability to identify all relevant instances, crucial in applications where missing a positive instance is costly.

F1-Score

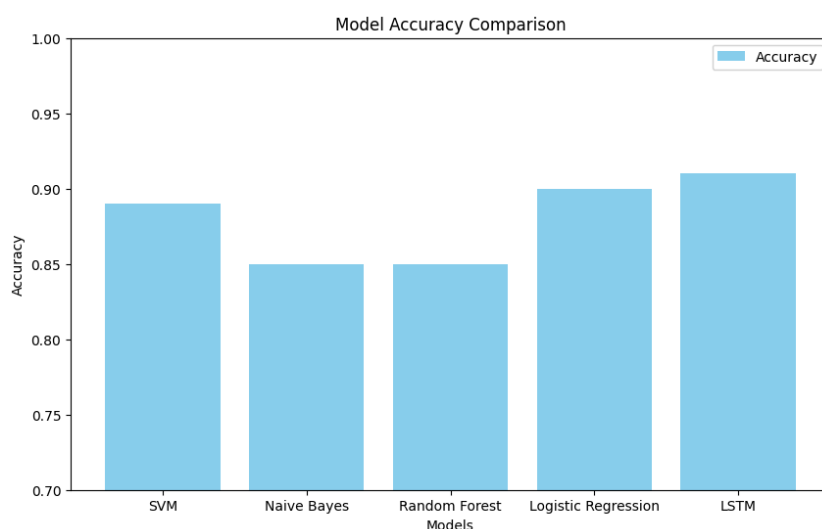
- **Definition:** The harmonic mean of precision and recall, balancing both metrics.
- **Formula:**

$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Purpose:** Useful when there is an imbalance between classes and a trade-off between precision and recall is necessary.

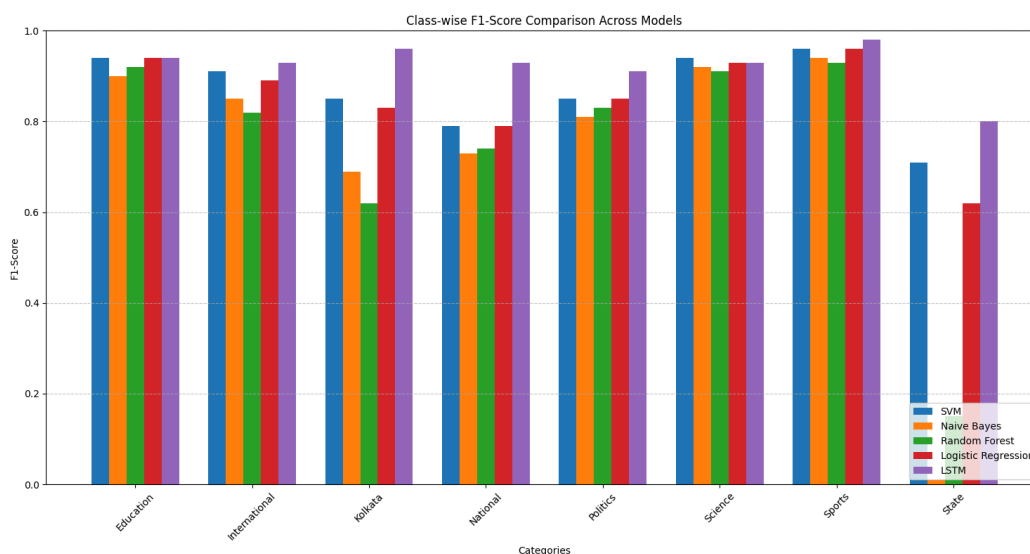
Overall Performance

- **LSTM** achieved the highest overall accuracy of **91%**, demonstrating its ability to capture complex patterns in text data. It also excelled in F1-scores across most categories.
- **Logistic Regression** and **SVM** followed with accuracies of **90%** and **89%**, respectively, offering strong and balanced performances.
- **Naive Bayes** and **Random Forest** performed similarly, achieving accuracies of **85%**, but struggled in handling imbalanced and small categories like "State."



Class-wise Observations

- **Education, Science, and Sports:** These categories showed consistently high F1-scores (above 90%) across all models, indicating they are easier to classify due to clear distinguishing features.
- **State:** A challenging category for all models due to its small dataset size and likely overlap with other categories. Naive Bayes performed poorly (F1-score of 0.01), while LSTM handled it best (F1-score of 0.80).
- **Kolkata and National:** These categories showed moderate performance across models, with LSTM and SVM handling them better than Naive Bayes and Random Forest.



Future Scope for Bengali News Classification Project

1. Addressing Class Imbalance:

Enhancing the model's performance for under-represented categories like **State** and **Kolkata** is crucial. Techniques such as Synthetic Minority Oversampling Technique (SMOTE), data augmentation, or weighted loss functions can be employed to balance the dataset and improve classification outcomes for minority classes.

2. Integration of Advanced Deep Learning Models:

Leveraging Bengali-specific pre-trained language models like **BanglaBERT** or fine-tuned transformer-based models like **mBERT** can significantly improve contextual understanding and classification accuracy. These models can handle the complexities of Bengali grammar and semantics more effectively than traditional machine learning algorithms.

3. Real-Time Classification System:

Developing a real-time news classification system would enable instant categorization of news articles as they are published. This involves integrating the trained model into a web or mobile application with APIs for seamless deployment, benefiting news agencies and content management systems.

4. Multi-label Classification Capability:

Extending the system to support multi-label classification will allow an article to be tagged with multiple relevant categories, such as **Politics** and **National**, which often overlap. This enhancement will make the classification system more flexible and applicable to real-world news data.

5. Incorporating Sentiment Analysis:

Adding sentiment analysis to the system can provide additional insights into the tone and emotional undertone of news articles, such as whether an article conveys positive, neutral, or negative sentiments. This feature can help readers and organizations better understand the intent behind news content and track public opinion trends.

Technologies Used

Editors Used to Write Code

- Jupyter Notebook
- Kaggle Notebook Editor
- Google Colab

Tools Used for Dataset Creation

- Microsoft Excel
- Google Spreadsheet

Programming Language and Libraries

The code is written in Python, utilizing the following libraries:

- **Data Manipulation and Preprocessing:**
 - Pandas
 - Numpy
 - NLTK
 - Sklearn
- **Web Scraping:**
 - BeautifulSoup
- **Visualization:**
 - Matplotlib
 - Seaborn
 - Wordcloud
- **Machine Learning and Neural Networks:**
 - Tensorflow
 - Keras

Conclusion

This project successfully built a machine learning system to classify Bengali news articles into distinct categories using datasets from prominent Bengali news sources. Models like Logistic Regression, Random Forest, SVM, Naïve Bayes, and LSTM were implemented, achieving high accuracy for well-represented categories such as **Education**, **Science**, and **Sports**.





Despite strong overall performance, challenges remain in accurately classifying under-represented categories like **State** and **Kolkata** due to class imbalance. Effective preprocessing and feature extraction techniques, including TF-IDF and n-grams, played a key role in capturing the nuances of Bengali text.

This work provides a strong foundation for Bengali NLP tasks, with scope for future improvements in handling imbalanced data, integrating deep learning models, and enabling real-time classification.

References

1. **Datasets:**
 - Dialect AI Shironaam Dataset: <https://huggingface.co/datasets/dialect-ai/shironaam>
 - Classification Bengali News Articles - IndicNLP: <https://www.kaggle.com/datasets/csoham/classification-bengali-news-articles-indicnlp>
 - Potrika Bangla Newspaper Dataset: <https://www.kaggle.com/datasets/sabbirhossainujjal/potrika-bangla-newspaper-datasets>
2. **Text Preprocessing:**
 - Bengali Stop Words List (GitHub): <https://github.com/stopwords-iso/stopwords-bn>
3. **Algorithms and Techniques:**
 - Scikit-learn for Machine Learning Models and Feature Engineering: <https://scikit-learn.org/>
 - TensorFlow/Keras for Deep Learning Models: <https://www.tensorflow.org/>
4. **Natural Language Processing:**
 - NLTK Library for Text Preprocessing: <https://www.nltk.org/>
 - TF-IDF and CountVectorizer: Scikit-learn Documentation
5. **Related Research:**
 - IndicNLP Library for Indian Languages: https://github.com/anoopkunchukuttan/indic_nlp_library
 - Applications of TF-IDF in NLP Tasks: Academic Literature and Research Papers
6. **General Tools and Libraries:**
 - Pandas and NumPy for Data Manipulation
 - Matplotlib and Seaborn for Data Visualization
 - Jupyter Notebooks for Experimentation and Prototyping

Project Repository

MCAHITK-MINOR-PROJECT/ **bangla_news_classifier**

A Machine Learning Classification Task to classify bengali news article into various categories

STARS
2