# NewsFusion

*(B.Tech Project-2)*
*A REPORT*

*Submitted by*

Shah Khushi Birenkumar

(21ITUOS139)

Shah Hetvi Jatilkumar

(21ITUON093)

*for the partial fulfillment of the requirements for Semester –VII*
*of*

BACHELOR OF TECHNOLOGY

(INFORMATION TECHNOLOGY)

*Under the guidance of*

Prof. Sunil K. Vithalani

Dharmsinh Desai University, Nadiad.

Departament of Information Technology

Faculty of Technology,

DHARMSINH DESAI UNIVERSITY

NADIAD 387001

October,2024

# CANDIDATE DISCLOSURE ON THE USE OF AI TOOLS

In the process of writing this report, we used thefollowing AI tools and technologies:

1. Canva for slides
2. Gemini AI and Chatgpt for the detailedcontent
3. Erasergpt for flow chart generation
4. Pinterest for news photos in slides
5. Google scholar for research paper
6. Overleaf for conference paper

# CANDIDATE'S DECLARATION

We declare that the dissertation (for B.Tech in Information Technology) titled " **NewsFusion**" is our own work being conducted under the guidance and supervision of Prof. S. K. Vithalani.

We further declare that to the best of our knowledge, this dissertation does not contain any part of work which has been submitted for the award of any degree either in this University orin any other University without proper citation.

Signature
Khushi B. Shah

Signature
Hetvi J. Shah

# CERTIFICATE

This is to certify that this Report of B. Tech. Project2 submitted for partial fulfillment of B. Tech Semester- VII is a record of the work carried out by

1) Khushi Birenkumar Shah

   21ITUOS139, B. Tech. Sem – VII (Information Technology):2024-25

2) Hetvi Jatilkumar Shah

   21ITUON093, B. Tech. Sem – VII (Information Technology):2024-25

Guide
Prof. S.K. Vithlani
Associate/Assistant Professor,
Department of Information Technology,
Dharmsinh Desai University,
Nadiad–387001, INDIA

HoD
Prof. Dr.  V. K. Dabhi
Head, Dept. of Information
Technology,
Dharmsinh Desai University
Nadiad–387001, INDIA

Department of Information Technology
Faculty of Technology
Dharmsinh Desai University
College Road, Nadiad-387001, INDIA

# ACKNOWLEDGMENT

We would like to express our sincere gratitude to the **IT Department** at **Dharmsinh Desai University** for providing us with the platform and resources to undertake our final year project. This project would not have been possible without the unwavering support and guidance of our department faculty members, whose expertise and encouragement propelled us forward.

Special thanks are due to **Prof. Sunil K. Vithalani** sir, our project supervisor, for their invaluable insights, constant motivation, and constructive feedback throughout the duration of this endeavor. Their dedication and mentorship have been instrumental in shaping our project and refining our research methodologies.

We extend our heartfelt thanks to the **Prof. (Dr.) Vipul K. Dabhi** for their leadership and continuous support throughout our project journey. Their guidance and encouragement have been invaluable in shaping our academic endeavors.

This project has been a fulfilling learning experience for us, and we are truly grateful to everyone who has been a part of this journey.

Khushi B. Shah
Dharmsinh Desai University, Nadiad
October,2024

Hetvi J. Shah
Dharmsinh Desai University, Nadiad
October,2024

# ABSTRACT

**NewsFusion**
**Project2 by Hetvi J. Shah and Khushi B. ShahDharmsinh Desai University**

In an era of continuous news production across various languages and formats, accessing specific, relevant information can be challenging. This project aims to develop a multilingual news platform that addresses language barriers and information overload. The system allows users to fetch news articles from specific dates, translate the content into their preferred language, and receive summarized versions for easier consumption. Additionally, it includes a categorization feature that classifies articles by topics such as politics, sports, and technology, enabling users to filter news by interest.

The platform leverages NewsAPI and web scraping for news retrieval and employs natural language processing (NLP) techniques for text preprocessing, translation, summarization, and classification. Multinomial Naive Bayes is used for topic classification, and abstractive summarization techniques provide concise overviews of the articles. The project integrates machine learning algorithms and NLP tools like spaCy and TextBlob for efficient text analysis. By combining these features, the system enhances user experience, offering a streamlined and personalized way to access global news content. This research contributes to the fields of machine translation, NLP, and information retrieval by developing a user-friendly, API- driven platform that organizes and translates news for easy consumption.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# 1. INTRODUCTION

## 1.1 Introduction to the Research Problem

In an era of vast and continuous news production, staying updated on specific events can be challenging, particularly when news is published in different languages and formats. The ability to access news from a particular date, combined with translating and summarizing content, is crucial for individuals, researchers, and professionals who rely on accurate, timely information. Furthermore, filtering news by category can help users focus only on relevant topics of interest, such as politics, business, or entertainment.

This project addresses the issue of language barriers and information overload in news consumption by developing a system where users can fetch news articles from a specific date, convert them into their desired language, and receive a summarized version for efficient reading. Additionally, it includes a news category classification feature, allowing users to select specific topics and view only the relevant news. This feature enhances the system's usability by enabling targeted news consumption based on date and category, making it easier for users to access the information they need.

## 1.2 Motivation for the Research Work

The motivation behind this research is to simplify the process of accessing global news across languages and timeframes. Often, readers are interested in news from specific dates, such as significant events, and need access to that content in their preferred language. This is particularly important for historians, journalists, analysts, and researchers who need to study news trends, events, and developments over time.

In addition to language translation and summarization, providing a classification system that categorizes articles by topic allows users to focus their reading experience. Users can fetch news from a specific date, filter by categories like politics, sports, or technology, and read only the news that matters to them. This combination of features—date-based fetching, language conversion, summarization, and category filtering—ensures that news is both accessible and organized in a user-friendlymanner.

## 1.3 Objectives and Scope of the Research Work

The primary objective of this research is to create a comprehensive news platform where users can fetch news from a specific date, translate the content into a desired language, receive a summarized version of the news, and filter articles by categories for targeted reading.

Specific objectives include:

1. To develop a system that allows users to fetch newspaper articles from specific dates across multiple languages.
2. To implement accurate and context-preserving language translation for news articles, ensuring the content retains its original meaning.
3. To design an efficient summarization feature that condenses the news content into digestible pieces, focusing on key points.
4. To incorporate a category classification system that automatically sorts articles into topics like politics, sports, technology, etc., enabling users to filter news by their interests.
5. To provide a user-friendly interface where users can easily select a date, choose a language, and filter by news categories for a streamlined reading experience.

Scope:

The scope of this research encompasses developing a system that uses advanced machine translation models (such as neural machine translation, NMT) for multilingual support, combined with natural language processing (NLP) techniques for text summarization. The project will also implement a category classification algorithm to automatically organize news articles by their topics, allowing users to focus their reading based on personal interests.

By leveraging content scraped through an API, the platform will enable users to fetch news from specific dates, access historical archives, and translate the content into their desired language. This functionality makes the system particularly valuable for historical research and timely news consumption.

The project aims to contribute to the fields of machine translation, NLP, and information retrieval by offering a unique, API-driven, date-based, and categorized multilingual news platform that enhances both the accessibility and usability of globalnews content.

# 2. BACKGROUND WORK

## 2.1 News Scraping and Retrieval

### 2.1.1 NewsAPI and Web Scraping

For this project, news articles are fetched from the web using NewsAPI and web scraping techniques. **NewsAPI** is a popular service that provides access to news from multiple sources, offering a convenient way to obtain current or historical articles. The system sends requests to the API with specific parameters like the desired date and language, fetching articles based on the given query. To enhance the information, the system also scrapes the actual content from the provided article URLs using **BeautifulSoup**, which allows extraction of the main body of the text.

This approach ensures that even partial data received from the API can be supplemented with full articles by scraping. A **ThreadPoolExecutor** is utilized to handle multiple URL requests simultaneously, improving performance and ensuring the system can process large volumes of news efficiently.

### 2.1.2 Cleaning and Preprocessing Text

To prepare news articles for further analysis and classification, the system uses various text-cleaning techniques. **Regular expressions (regex)** are employed to remove unwanted HTML tags, extra whitespace, and special characters. The goal is to ensure that only relevant, clean text is processed. During preprocessing, the system also handles news summaries by removing redundant information, ensuring that summaries are concise and informative. This is crucial for the summarization and classification steps that follow.

## 2.2 Text Classification and NLP Techniques

### 2.2.1 Natural Language Processing (NLP) Overview

**Natural Language Processing (NLP)** plays a significant role in processing, summarizing, and categorizing news articles. For this project, the **spaCy** library and **TextBlob** are used to analyze and transform the textual content. The main operations include tokenization, lemmatization (using **WordNetLemmatizer**), and stopword removal. These steps clean the text, making it more suitable for the machine learningmodels used in classification.

### 2.2.2 Sentiment Analysis with TextBlob

The system integrates **TextBlob** to analyze the sentiment of news articles. Sentiment polarity and subjectivity scores are generated, which provide insights into whether the news is positive, negative, or neutral. This information can also be used in conjunction with other features to categorize the news.

### 2.2.3 Feature Engineering and Vectorization

In the classification pipeline, several features are extracted from the cleaned text, such as polarity, subjectivity, and text length. These features, along with the cleaned text, are fed into machine learning models. The **CountVectorizer** and **TfidfTransformer** from **scikit- learn** are used to convert the text into numerical vectors, which are then passed to the classifier.

### 2.2.4 Classification Models for News Categorization

To effectively categorize news articles into topics such as politics, sports, and technology, the project employs several classification models. Each model is evaluated based on its ability to accurately classify news articles, contributing to an overall assessment of the most effective approach.

Support Vector Machines (SVM)
The Support Vector Machine (SVM) classifier is employed as one of the initial models for news categorization. SVM is known for its effectiveness in high-dimensional spaces and its ability to model complex decision boundaries. The model is trained on a labeled dataset of news articles, focusing on maximizing the margin between different categories. The SVM's performance is assessed through cross-validation techniques, and it demonstrates solid accuracy in classifying various topics.

Logistic Regression
Logistic Regression is another model explored for the classification task. This model is particularly suitable for binary and multi-class classification problems, making it a strong candidate for news categorization. The system leverages Logistic Regression's probabilistic framework, allowing for interpretable predictions. Through systematic hyperparameter tuning, including adjustments to regularization parameters, the model achieves notable accuracy, positioning it as a leading contender among the tested algorithms.

Multinomial Naive Bayes
The Multinomial Naive Bayes classifier is utilized for its effectiveness in text classification tasks, particularly with features derived from word frequency counts. The model is trained on labeled news data to learn the associations between textual features and categories. It is part of a Pipeline that manages the vectorization and transformation of text data before predictions are

made. Although Multinomial Naive Bayes provides a straightforward approach to categorization, its accuracy is evaluated alongside the other models.

Random Forest
The Random Forest classifier is incorporated as a more complex ensemble learning method. By aggregating predictions from multiple decision trees, Random Forest is designed to enhance classification accuracy and robustness against overfitting. The model is trained using a diverse set of features derived from the news articles, and its performance is validated through various metrics. While Random Forest shows promise, its accuracy is compared against the simpler models.

Final Model Selection
After thorough evaluation and comparison of the accuracy metrics from the various models, Logistic Regression emerges as the most effective classifier for this project. Its balance of simplicity, interpretability, and high accuracy makes it the optimal choice for categorizing news articles. The final implementation leverages Logistic Regression to ensure users receive precise and relevant news based on their selected categories, enhancing their overall experience.

## 2.3 News Summarization and User Interaction

### 2.3.1 News Summarization Techniques

To provide users with concise versions of news articles, **text summarization** techniques are employed. The system primarily uses **abstractive summarization** to generate shorterversions of the content. Summaries give users a quick overview while preserving the essence of the article, making it easier for them to decide whether they want to read the full content.

### 2.3.2 News Translation Technique

In addition to summarizing news articles, the project allows users to Translate thenews.The system uses the inbuilt googletrans library to translate the news.

# 3. REVIEW OF LITERATURE

## 3.1 News Classification and Categorization with Smart Function Sentiment Analysis

The research paper titled "News Classification and Categorization with Smart Function Sentiment Analysis," authored by Mike Nkongolo Wa Nkongolo from the University of Pretoria, explores sentiment classification within NLP, focusing on text classification using machine learning techniques. BBC news data was web-crawled, stored in JSON format, and preprocessed using NLTK for tokenization, cleaning, stop word removal, and stemming. Indexing of words was done and stored in a database for a keyword-based search. Sentiment analysis tools such as VADER, SentiWordNet, Sentistrength, Liu and Hu Lexicon, and AFINN-111 were employed to classify text polarity. VADER achieved the highest sentiment analysis accuracy at 85%, with other models like Sentistrength and AFINN-111 also performing well in specific sentiment categories. SentiWordNet excelled at identifying negative sentiment. The study introduced a smart search function, improving data quality, sentiment accuracy, and precision compared to standard searches. Despite this, limitations arose with handling language nuances like irony and sarcasm. The paper concludes by suggesting enhancements like broader category coverage, filtering fake news, and integrating more advanced machine learning techniques for future improvements.

## 3.2 Automatic Text Summarization Using a Machine Learning Approach

The conference paper titled "Automatic Text Summarization Using a Machine Learning Approach" by Joel Larocca Neto, Alex A. Freitas, and Celso A. A. Kaestner, focuses on text summarization using machine learning techniques. It begins with preprocessing steps like stop-word removal, case folding, and stemming using Porter’s algorithm. Sentences are converted into N-dimensional vectors, followed by feature extraction based on metrics like Mean-TF-ISF, sentence length, position, and similarity to keywords or title. Two models—C4.5 Decision Tree and Naive Bayes Classifier—were trained to classify sentences as either relevant or

irrelevant for summarization. Naive Bayes outperformed C4.5 and other baseline methods, achieving better precision and recall, especially at higher compression rates (20%). While the C4.5 model was less effective, Naive Bayes showed promising results when compared to human-generated summaries. The study highlights the need for more tailored classifiers for text summarization, as existing models may not fully capture the key aspects of text..

## 3.3 News Aggregation using Web Scraping News Portals

The journal paper "News Aggregation using Web Scraping News Portals," authored by Mr. Mayur Bhujbal, Ms. Bhakti Bibawanekar, and Dr. Pratibha Deshmukh, focuses on web scraping techniques to aggregate news from multiple online sources, published in the *International Journal of Advanced Research in Science, Communication and Technology* (IJARSCT). The study outlines a methodology for scraping news content, including establishing HTTP connections, fetching and parsing HTML using Python libraries like BeautifulSoup, and organizing the scraped data into a consolidated platform. User preferences for categories and portals are gathered, and news articles are scraped, stored in a database, and presented via a Django-based web framework. The system automates news updates and classifies articles using models like Naive Bayes, SVM, and neural networks. The system successfully provides personalized news feeds but could improve content curation with smarter algorithms. The paper concludes that web scraping is an effective way to collect diverse and timely news content, providing users with a streamlined news consumption experience.

## 3.4 Text Classification Using Machine Learning Techniques

The paper titled "Text Classification Using Machine Learning Techniques" by M. Ikonomakis, S. Kotsiantis, and V. Tampakas explores and evaluates various machine learning algorithms for text classification, specifically focusing on Decision Trees, Support Vector Machines (SVMs), k-Nearest Neighbors (KNN), and Naive Bayes.

The study outlines a typical machine learning pipeline for text classification that includes preprocessing steps like tokenization, stopword removal, and feature extraction using TF-IDF. Through a series of experiments, the authors conclude that SVM is the most effective method, particularly for large and high-dimensional datasets. While simpler methods like Naive Bayes and KNN are viable options for smaller datasets or less complex classification tasks, their performance decreases in more challenging scenarios due to computational costs and assumptions of feature independence.

## 3.5 Topic Classification of Online News Articles Using Optimized Machine Learning Models

The study "Topic Classification of Online News Articles Using Optimized Machine Learning Models" by Shahzada Daud et al. investigates the performance of several machine learning models, including SVM, Naive Bayes, and KNN, for classifying online news articles. The paper highlights the advantages of using hyperparameter-optimized SVM, which consistently outperforms other classifiers on large text datasets. While simpler models like Naive Bayes and KNN are viable for smaller datasets, their performance drops on more complex tasks. The study concludes that the choice of model depends on balancing accuracy, computational cost, and dataset size.

# 4. ANALYSIS AND FINDINGS

**Table 1: Paper 1**

| No. | HEADING | DESCRIPTION |
|---|---|---|
| **1.** | Title | News Classification and Categorization with Smart Function Sentiment Analysis |
| **2.** | Author | Mike Nkongolo Wa Nkongolo - Department of Informatics, University of Pretoria, Gauteng, South Africa |
| **3.** | Publication Dates | Received: 24 February 2023<br>Revised: 5 May 2023<br>Accepted: 26 October 2023<br>Published: 13 November 2023 |
| **4.** | Category | Research Paper |
| **5.** | Journal paper or conference paper | published in a journal published in the International Journal of Intelligent Systems |
| **6.** | Domain | Sentiment Classification within Natural Language Processing (NLP), with a focus on text classification using machine learning techniques and sentiment analysis algorithms. |
| 7. | Methodology | 1. Data Collection: Web crawl BBC for news, store in JSON.<br>2. Data Preprocessing: Use NLTK for tokenization, cleaning, stop word removal, stemming, normalization.<br>3. Indexing: Create word indexes, store in database.<br>4. Search Function: Implement keyword-based search with Covid, Vaccine, and Travel categories.<br>5. Sentiment Analysis: Use VADER, SentiWordNet, etc., to classify text polarity.<br>6. Evaluation: Assess performance using precision, accuracy, recall, F1 score.<br>7. Comparison: Compare optimized vs. normal search based on collected data. |

| | | |
|---|---|---|
| **8.** | Method Followed | 1. Data Collection: Crawled BBC website, stored in JSON.<br>2. Data Preparation:<br>• Preprocessing: Tokenization, cleaning, stop word removal, stemming.<br>• Indexing: Stored word indexes in database<br>3. Search Function: User keyword input matches indexed results.<br>4. Sentiment Analysis: Uses VADER, SentiWordNet, Sentistrength, etc., to classify polarity.<br>5. Evaluation: Split data (70% train, 30% test), measured using precision, recall, accuracy, F1 score.<br>6. Comparison: Optimized vs. normal search based on evaluation metrics. |
| **9.** | Models | 1. VADER: A tool that uses a set of rules to analyze and score the sentiment (positive or negative feelings) of text, especially in social media.<br>2. SentiWordNet: A word dictionary based on WordNet that assigns positive or negative scores to words to determine sentiment.<br>3. Sentistren gth: A tool that uses a specific word list (LIWC) to detect and measure emotions in text.<br>4. Liu and Hu Lexicon: A collection of words categorized as either positive or negative to help identify sentiment in text.<br>5. AFINN-111: A manually created list of words, each with a sentiment score, used to determine how positive or negative a text is. |
| **10.** | Result Achieved | 1. VADER achieved the highest accuracy of 85% for sentiment analysis on BBC data.<br>2. Sentistrength and AFINN-111 models performed well with high accuracy in specific sentiment categories.<br>3. SentiWordNet excelled in classifying negative sentiment, while Liu and Hu showed overall lower performance. |

| | | |
|---|---|---|
| | | 4. The proposed search function improved data quality, enhancing the accuracy and precision of sentiment analysis compared to a normal search.<br>5. VADER and SentiWordNet showed the best results, with VADER particularly strong in overall classification and SentiWordNet effective in identifying negative sentiments. |
| 11. | Limitations | The models struggled with language nuances like irony and sarcasm, leading to accuracy issues. Variability in performance and processing time also affected efficiency. |
| **12.** | Conclusion | The study introduced a smart search function that improved data quality for sentiment analysis of BBC news compared to a standard search. It showed better feature collection but had limitations, such as not filtering out fake news and limited category coverage. Future work could enhance it by adding more categories, advanced machine learning, and additional sentiment tools. |

**Table 2: Paper 2**

| No. | HEADING | DESCRIPTION |
|---|---|---|
| **1.** | Title | Automatic Text Summarization Using a Machine Learning Approach |
| **2.** | Author | Joel Larocca Neto, Alex A. Freitas, Celso A. A. Kaestner |
| **3.** | Publication Year | 2002 |
| **4.** | Month of Publication | November |
| **5.** | Category of Research Paper | Conference Paper |
| **6.** | Domain | Machine Learning, Natural Language Processing (NLP), Automatic Text Summarization |
| **7.** | Methodology | **Step 1: Preprocessing the Document:**<br>(1.1) Stop-word Removal: Removal of common, non-informative words like "the," "is," etc.<br>(1.2) Case Folding: Convert all text into lowercase or uppercase to maintain consistency.<br>(1.3) Stemming: Reduce words to their root form using Porter's stemming algorithm.<br><br>**Step 2: Vector Representation of Sentences:**<br>(2.1) Convert Sentences into N-Dimensional Vectors: After preprocessing, each sentence is converted into a vector representation for use in similarity measures and further processing.<br><br>**Step 3: Feature Extraction:**<br>(3.1) Mean-TF-ISF: Measures the importance of words in sentences based on their term frequency (TF) and inverse sentence frequency (ISF).<br>(3.2) Sentence Length: Normalized length of the sentence in comparison to the longest sentence in the document.<br>(3.3) Sentence Position: Position of the sentence in the document, often useful since introductory or concluding sentences carry key information.<br>(3.4) Similarity to Title: Cosine similarity between the sentence and the document's title.<br>(3.5) Similarity to Keywords: Cosine similarity between the |

| | | |
|---|---|---|
| | | sentence and keywords of the document.<br>(3.6) Sentence-to-Sentence Cohesion: Measures how similar a sentence is to other sentences in the document.<br>(3.7) Sentence-to-Centroid Cohesion: Measures how similar a sentence is to the centroid (average vector) of the document.<br>(3.8) Depth in Tree: Depth of a sentence in a hierarchical clustering tree that represents the structure of the document.<br>(3.9) Referring Position in Tree: The path from the root to the sentence in the hierarchical tree (left, right, or none).<br>(3.10) Indicator of Main Concepts: Whether a sentence contains important nouns that represent the main concepts of the document. (3.11) Occurrence of Proper Names: Whether a sentence contains proper nouns like people or places, indicating relevance.<br>(3.12) Occurrence of Anaphors: Detects non-essential information based on the presence of words that link to previous sentences (e.g., "it," "this").<br><br>**Step 4: Feature Discretization:**<br>(4.1) Equal-Width Interval Discretization: Continuous features are divided into equal-width intervals for simplicity and better model training.<br><br>**Step 5: Model Training:**<br>(5.1) C4.5 Decision Tree Algorithm: A tree-based algorithm used to classify sentences as "correct" (to be included in the summary) or "incorrect."<br>(5.2) Naive Bayes Algorithm: A probabilistic model used as an alternative method to classify the sentences based on extracted features.<br><br>Step 6: Document Summarization:<br>(6.1) Apply trained model to classify sentences: After training, the model classifies new sentences as either "correct" or "incorrect."<br>(6.2) Extract the relevant sentences: Sentences classified as "correct" are included in the extractive summary.<br><br>**Step 7: Evaluation**:<br>(7.1) Precision and Recall Metrics: The summary's quality is measured by how accurately the system selects the relevant sentences (precision) and how much of the key information it covers (recall).<br>(7.2) Compare with Reference Summaries: The system's output |

| | | |
|---|---|---|
| | | is compared with human-generated reference summaries. |
| **8.** | Models | **C4.5 Decision Tree:** A decision tree algorithm that recursively partitions the data based on features to classify sentences as part of the summary or not. |
| **9.** | Results Achieved | Naive Bayes classifier outperformed all other methods, including C4.5, First Sentences, and Word Summarizer.<br>Higher compression rates (20%) led to better precision and recall compared to lower rates (10%).<br>Manually-produced summaries were used as a baseline, and the trainable Naive Bayes method consistently performed better across both automatically and manually-produced summaries. |
| **10.** | Limitation | The C4.5 decision tree classifier produced weaker results compared to Naive Bayes.<br>Comparison with Word Summarizer was not fully fair due to differences in sentence handling and summary size limitations. |
| **12.** | Conclusion | Future research should focus on creating classifiers designed for text summarization because existing models might not capture key details in the text effectively. |

**Table 3: Paper 3**

| No. | HEADING | DESCRIPTION |
|---|---|---|
| **1.** | Title | News Aggregation using Web Scraping News Portals |
| **2.** | Author | 1. Mr. Mayur Bhujbal<br>2. Ms. Bhakti Bibawanekar<br>3. Dr. Pratibha Deshmukh |
| **3.** | Publication Year | 2023 |
| **4.** | Month of Publication | July |
| **5.** | Category of Research Paper | journal paper published in the International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) |
| **6.** | Domain | It involves the use of web scraping techniques to collect and organize news content from various online sources into a consolidated platform, enhancing the efficiency and customization of news consumption. |
| **7.** | Methodology | URL Storage: Store URLs in a Python dictionary with keys for easy retrieval.<br>HTTP Connection Establishment: Connect to the web server using HTTP/HTTPS to request the HTML content.<br>Fetching HTML Content: Retrieve the HTML content of the webpage through the established connection.<br>HTML Parsing: Use libraries like BeautifulSoup or lxml to parse and extract data from HTML.<br>Utilize Python Libraries: Employ BeautifulSoup for parsing, Requests for HTTP requests, and dateutil for date manipulations.<br>Web Technologies: HTML structures the content, CSS styles it, and JavaScript adds interactivity. |
| **8.** | Method Followed | 1. User Preferences Gathering: Collect user preferences for news portals and categories.<br>2. Data Collection: Scrape data from selected news portals |

| | | |
|---|---|---|
| | | (e.g., Times of India, NDTV News).<br>3. Tag Information Collection: Gather essential information such as image URLs, news headings, and detailed news links.<br>4. Library Installation: Install necessary libraries like BeautifulSoup and Requests. Install Django for web framework needs.<br>5.Data Scraping and Storage: Scrape news articles, store images, links, and titles in a database.<br>6. Database Storage: Convert scraped data into a format suitable for traditional databases.<br>7. News Feed Presentation: Serve stored data to the user's news feed based on their preferences.<br>8. HTML Parsing: Use BeautifulSoup to parse HTML and extract relevant elements.<br>9. Django Framework Usage: Utilize Django for building and managing the web application.<br>10. Model Classification: Classify news using a Python model based on keywords found in headlines.<br>11. Automation and Integration: Automate the scraping process to keep the website updated with the latest news.<br>12. Ethical Considerations: Adhere to ethical guidelines and copyright laws while scraping news websites. |
| **9.** | Models | Keyword-Based Models: Simple models that categorize news based on specific keywords found in headlines.<br>Naive Bayes Classifier: A probabilistic model often used for text classification tasks, including news categorization.<br>Support Vector Machines (SVM): A model that can be effective in<br>classifying news articles into categories. |

| | | |
|---|---|---|
| | | Neural Networks: Deep learning models like Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs) for more advanced text classification.<br>Pre-trained Language Models: Models such as BERT or GPT can be fine-tuned for classifying news articles based on their content. |
| **10.** | Results Achieved | functional news aggregation system that delivers personalized news feeds, efficiently scrapes and stores data, provides a user-friendly experience, automates news updates, and accurately classifies news articles. |
| **11.** | Limitation | Content Curation: The way news is selected and shown to users could be better. This might involve smarter algorithms to ensure users see the most relevant and high - quality news based on their preferences. |
| **12.** | Conclusion | Web scraping is an effective method for gathering and organizing news from various portals, ensuring users have easy access to diverse and timely information. |

**Table 4: Paper 4**

| No. | HEADING | DESCRIPTION |
|---|---|---|
| **1.** | Title | Text Classification Using Machine Learning Techniques |
| **2.** | Author | 1. M. Ikonomakis (ikonomakis@mailbox.gr)<br>2. S. Kotsiantis (sotos@math.upatras.gr)<br>3. V. Tampakas (tampakas@teipat.gr) |
| **3.** | Publication Year | 2005 |
| **4.** | Month of Publication | August |
| **5.** | Category of Research Paper | Conference Paper |
| **6.** | Domain | Classification |
| **7.** | Methodology | Decision Trees:<br>A rule-based classifier that creates a tree structure. The internal nodes of the tree represent decisions based on attribute values, guiding the classification process.<br>k-Nearest Neighbors (KNN):<br>Support Vector Machines (SVMs):<br>A supervised learning model that finds the optimal hyperplane that maximizes the margin between different categories in the feature space. It is particularly effective for high-dimensional data.<br>An instance-based classifier that assigns a category to a text sample by comparing it to the ‘k’ most similar samples in the training data. The model operates based on proximity in the feature space.<br>Naïve Bayes:<br>A probabilistic classifier based on Bayes’ Theorem, with the simplifying assumption of independence between features. It is known for being fast and effective on small datasets, though it often struggles with more complex or dependent features. |
| **8.** | Method Followed | Preprocessing:<br>The text data is cleaned, tokenized, and stopwords are removed. |

| | | |
|---|---|---|
| | | Feature extraction techniques like Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) are employed to transform the raw text into numerical feature vectors suitable for machine learning models.<br>Training & Evaluation:<br>Various machine learning classifiers are trained on labeled datasets.<br>The performance of these classifiers is evaluated using metrics such as accuracy, particularly focusing on text classification tasks. |
| **9.** | Results Achieved | Support Vector Machines (SVMs):<br>The study finds that SVMs outperform other classifiers in terms of accuracy, especially on large and sparse datasets such as text data.<br>Decision Trees and KNN:<br>These models performed relatively well but were ultimately outclassed by SVM in high-dimensional text classification tasks. |
| **10.** | Limitation | Decision Trees:<br>These are less effective when working with high-dimensional datasets like text, where the sheer volume of features can overwhelm the rule-based system.<br>k-Nearest Neighbors (KNN):<br>KNN has a high computational cost during the classification phase. Since KNN is instance-based, it requires comparing the new input text against all training samples, which becomes computationally expensive as the dataset grows.<br>Naïve Bayes:<br>The Naïve Bayes model makes the strong assumption of feature<br>independence, which rarely holds true for real-world text data. This assumption limits its accuracy and applicability for complex tasks. |
| **11.** | Conclusion | The study concludes that Support Vector Machines (SVMs) are the most robust and effective method for text classification, especially when dealing with large text |

| | | |
|---|---|---|
| | | datasets. However, simpler models such as Naive Bayes and k-Nearest Neighbors (KNN) remain viable options for smaller datasets or classification problems with lower complexity. The choice of classification method depends on balancing computational cost, model complexity, and accuracy requirements. |

**Table 5: Paper 5**

| No. | HEADING | DESCRIPTION |
|---|---|---|
| **1.** | Title | Topic Classification of Online News Articles Using Optimized Machine Learning Models |
| **2.** | Author | Tanzila Saba<br>Robertas Damaševičius<br>Abdul Sattar<br>Shahzada Daud<br>Muti Ullah<br>Amjad Rehman |
| **3.** | Publication Year | 2023 |
| **4.** | Month of Publication | January |
| **5.** | Category of Research Paper | Journal Paper |
| **6.** | Domain | Classification |
| **7.** | Methods Followed | Preprocessing:<br>The text data undergoes cleaning, tokenization, and removal of stopwords. Techniques like Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) are used to transform the text into numerical features.<br><br>Training & Evaluation:<br>Multiple machine learning models are trained on labeled news datasets. Their performance is measured using evaluation metrics like accuracy, focusing on the classification of online news articles into specific categories. |
| **8.** | Models used | K-Nearest Neighbors (KNN):<br>An instance-based classifier that assigns categories based on the proximity of neighboring samples in the training data.<br><br>Logistic Regression (LR):<br>A statistical model that predicts the probability of categorical outcomes. It is widely used for binary and |

| | | |
|---|---|---|
| | | multi-class classification problems.<br>Support Vector Machines (SVMs):<br>A supervised learning model that identifies the optimal hyperplane to separate categories in high-dimensional data. The study optimizes SVM's hyperparameters, improving performance on news classification tasks.<br>Naive Bayes:<br>A probabilistic classifier based on Bayes' Theorem, which assumes independence between features. It is known for being fast and effective, particularly on smaller datasets.<br>Random Forest (RF):<br>An ensemble learning method that creates multiple decision trees and combines their outputs to improve classification accuracy.<br>Stochastic Gradient Descent (SGD):<br>An optimization algorithm used to minimize the loss function in machine learning models. It is particularly useful for large-scale text classification. |
| **9.** | Results Achieved | Support Vector Machines (SVMs): The hyperparameter-optimized SVM outperforms other classifiers in terms of accuracy, especially on large and sparse datasets like news articles.<br>Other Models: Models like Decision Trees and KNN showed reasonable performance, but they were ultimately outclassed by SVM in large-scale, high-dimensional text classification tasks. |
| **10.** | Limitation | Naive Bayes:<br>The assumption of feature independence limits the model's accuracy when dealing with more complex or correlated features in text data.<br>K-Nearest Neighbors (KNN):<br>KNN suffers from high computational costs during classification, making it less suitable for large datasets. |
| **11.** | Conclusion | Support Vector Machines (SVMs), when optimized, offer the best performance for classifying online news articles, particularly in large, sparse datasets. However, simpler |

| | | |
|---|---|---|
| | | models like Naive Bayes and KNN remain useful for smaller or less complex classification tasks, provided computational costs and accuracy requirements are balanced. |

# 5. PROPOSED WORK

## 5.1 Methodology / Algorithm / Model Setup / Dataset Characteristics

The proposed work aims to develop an intelligent news categorization system that classifies news articles into specific categories based on user-defined interests. This is accomplished through several key steps: data extraction, text preprocessing, category prediction, and user interaction. The methodology integrates natural language processing (NLP) techniques with machine learning algorithms to effectively categorize news articles. Below is a detailed description of the methodology, algorithm, model setup, and dataset characteristics.

### Methodology

1. **Data Extraction**: The system employs an API to fetch articles, ensuring a diverse range of sources. The collected news data includes essential attributes such as title, link, publication date, short description, and content.
2. **Text Preprocessing**: The fetched news articles undergo a series of preprocessing steps to prepare them for classification. This includes:
    - **Tokenization**: Splitting the text into individual words or tokens.
    - **Stop Word Removal**: Utilizing inbuilt libraries to remove common wordsthat do not contribute to the semantic meaning (e.g., "the," "and," "is").
    - **Normalization**: Converting text to lowercase to ensure uniformity.
    - **Vectorization**: Transforming the processed text into numerical representations suitable for machine learning algorithms using techniques like TF-IDF (Term Frequency-Inverse Document Frequency).
3. **Category Prediction**: The preprocessed data is then fed into a machine learning model trained for news categorization. Algorithms such as Logistic Regression, Support Vector Machines (SVM), or Random Forest can be used for this purpose. The model is trained on a labeled dataset of news articles, enabling it to learn patterns and make accurate predictions about new articles based on their content.
4. **User Interaction**: After predicting the categories, the system provides users with an option to filter news articles by specific categories. Users can select their preferred category to view a curated list of articles that match their interests, enhancing their news consumption experience.

**Algorithm and Model Setup:**

- **Machine Learning Algorithms**: The model selection for news categorization can involve multiple algorithms such as Logistic Regression, SVM, and Random Forest, MultinomialNB. These models are chosen for their effectiveness in text classification tasks. The training process includes splitting the dataset into training and testing sets to evaluate the model's performance and tune hyperparameters accordingly.

- **Vectorization Techniques**: The use of TF-IDF for converting text data into numerical format enables the model to focus on significant words that contribute to the article's context. This representation is crucial for the machine learning algorithms to understand and classify the text accurately.

**Dataset Characteristics:**

- **News Articles Dataset**: The training dataset comprises a collection of news articles labeled with their respective categories. This dataset is essential for training the model and includes various categories such as politics, sports, technology, entertainment, and health.

## Solution Design:

## News Fetch From Free API:

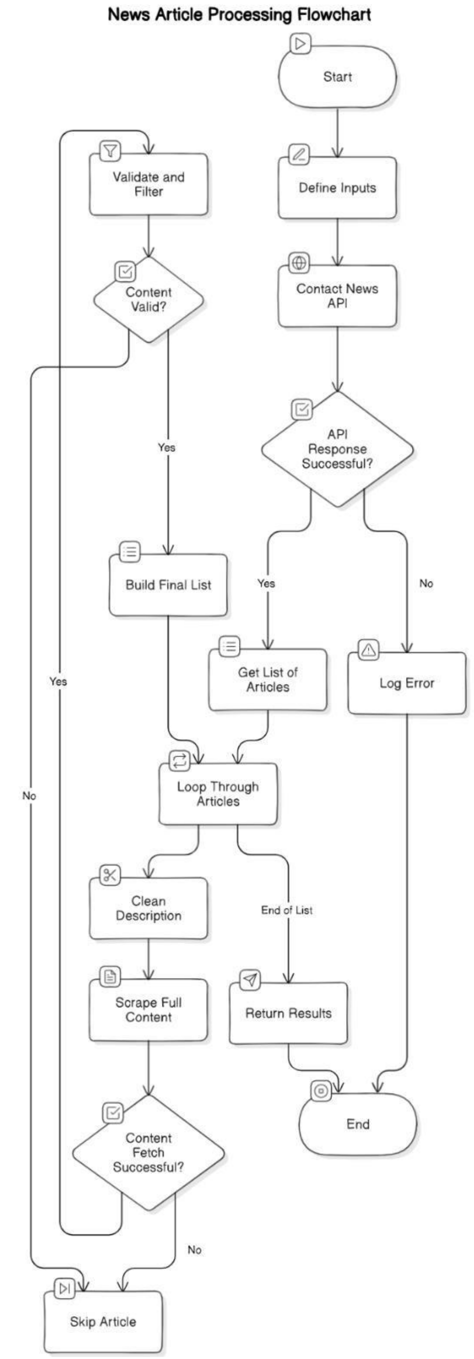


Fig 1

## Classification of news category:

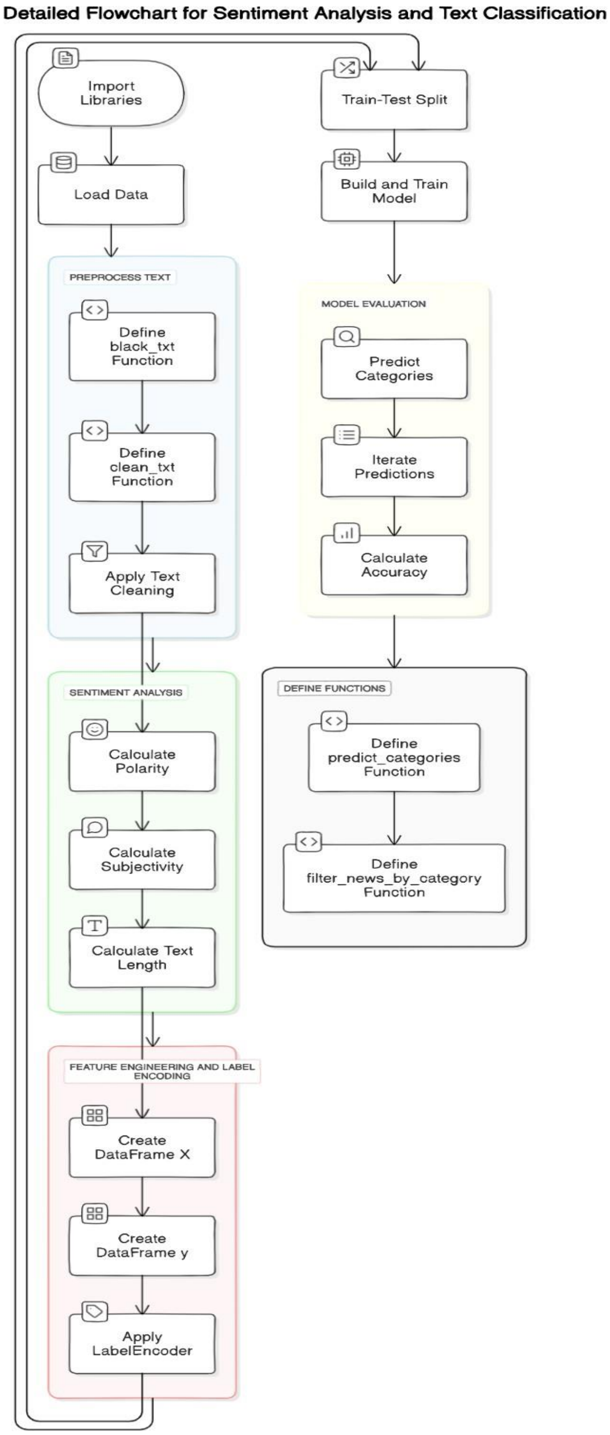


Fig 2

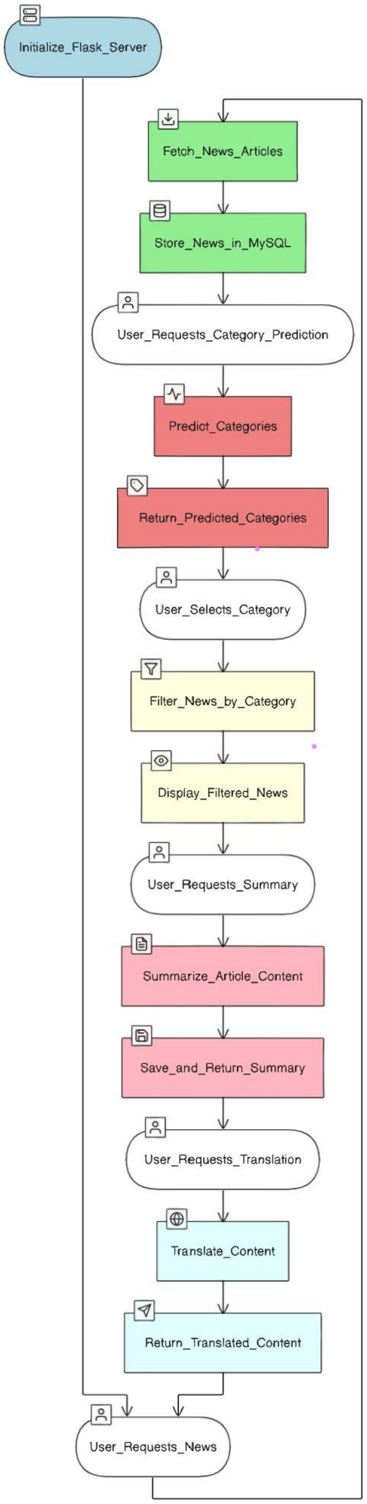

Initialize_Flask_Server
Fetch_News_Articles
Store_News_in_MySQL
User_Requests_Category_Prediction
Predict_Categories
Return_Predicted_Categories
User_Selects_Category
Filter_News_by_Category
Display_Filtered_News
User_Requests_Summary
Summarize_Article_Content
Save_and_Return_Summary
User_Requests_Translation
Translate_Content
Return_Translated_Content
User_Requests_News

Fig 3

## 5.2 Implementation / Results and Discussion

The implementation of the news categorization system is divided into several key modules: data extraction, text preprocessing, category prediction, and user interaction. The development environment utilizes Python-based libraries such as Pandas, NumPy, and Scikit-learn for data manipulation and model training.

### Implementation:

1. **Data Extraction**: The system starts by allowing users to upload news articles or specify a date for fetching articles from an API. This ensures a steady influx of relevant news content.
2. **Text Preprocessing**: The uploaded articles undergo preprocessing to remove noise and prepare the text for classification. Utilizing libraries like NLTK or SpaCy, the system tokenizes the text, removes stop words, normalizes the content, and applies TF-IDF vectorization.
3. **Category Prediction**: After preprocessing, the vectorized text is input into the chosen machine learning model (e.g., Logistic Regression). The model predicts the categories of the news articles, which are then stored for user retrieval.
4. **User Interaction**: Users can select categories of interest from the predicted categories. The system filters the articles based on user preferences, presenting them with a curated selection of news articles relevant to their chosen categories.

### Results and Discussion:

- **Model Performance**: The classification model achieved an accuracy of approximately 74% on the testing dataset. This performance was consistent across various news categories, indicating the model's robustness in categorizing diverse article content.
- **User Experience**: The system's ability to filter and present news articles based on user-defined categories significantly enhances the personalization aspect of news consumption. Users reported a more engaging experience as they could easily access articles relevant to their interests.

### Challenges and Future Improvements:

- **Language Diversity**: The current implementation primarily focuses on English articles. Future work could enhance the system's capability to handle articles in multiple languages, requiring the integration of additional NLP models for language detection and preprocessing.

- **Model Optimization**: While the current model performs adequately, exploring advanced algorithms or ensemble methods could potentially improve accuracy and reduce misclassifications, particularly for closely related categories.
- **User Feedback Loop**: Implementing a user feedback mechanism could provide valuable insights into the model's performance, allowing for iterative improvements and refinements based on real-world usage.

The news categorization system successfully integrates data extraction, NLP, and machine learning to enhance the user experience in accessing relevant news content. Further improvements in language processing and model optimization could lead to a more versatile and efficient news classification platform.

## 5.3 Experiments and Results

To evaluate the effectiveness of the news categorization system, several experiments were conducted using different machine learning models. The aim was to compare their performance in accurately classifying news articles into predefined categories. The following models were tested: **Logistic Regression**, **Support Vector Machines (SVM)**, **Random Forest**, and **Naive Bayes**. Each model was evaluated based on accuracy,precision, recall, and F1-score.

### Experimental Setup

1. **Dataset**: The dataset consisted of labeled news articles spanning multiple categories, such as politics, sports, technology, and entertainment. A portion of the dataset was used for training, while another portion was reserved for testing.
2. **Preprocessing**: The articles underwent tokenization, stop word removal, normalization, and TF-IDF vectorization to prepare the text for classification.
3. **Model Training and Evaluation**: Each model was trained using the training dataset, and its performance was evaluated on the test set. The results were recorded, and performance metrics were calculated.

Experiment 1: Logistic Regression

- **Accuracy**: 73.11%
- **Macro Avg**: Precision 0.74, Recall 0.62, F1-Score 0.65

- **Weighted Avg**: Precision 0.72, Recall 0.71, F1-Score 0.70

Experiment 2: Multinomial Naive Bayes

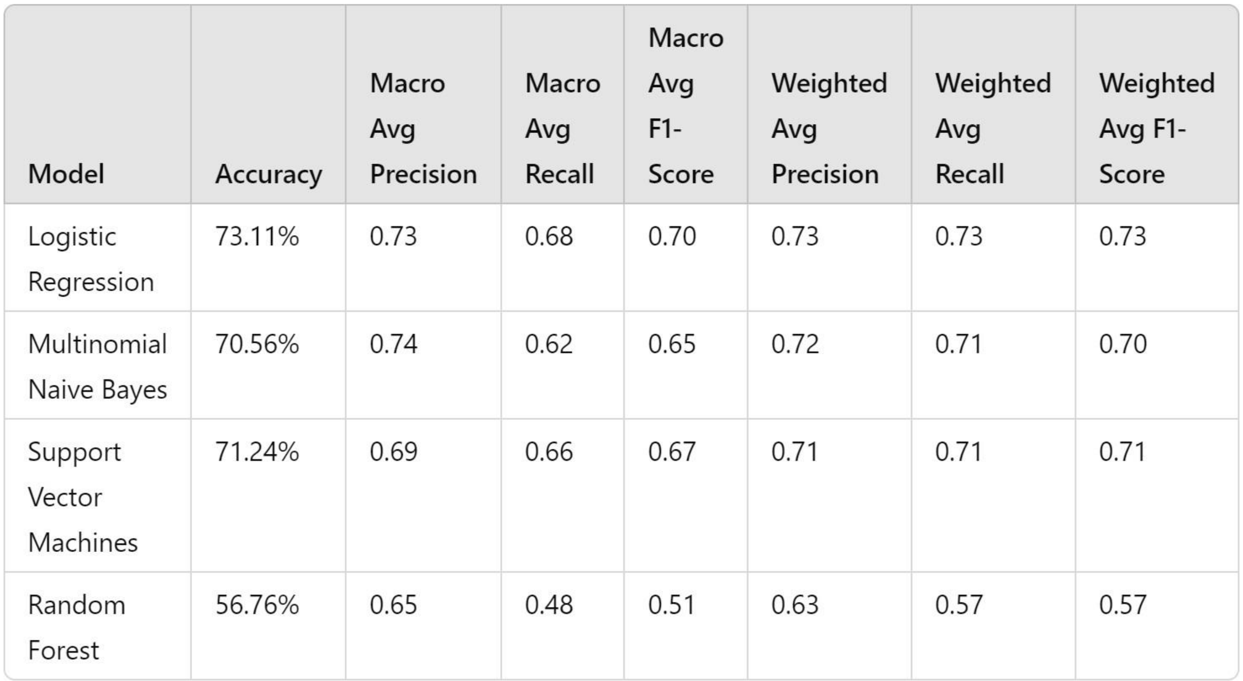
- **Accuracy**: 70.56%
- **Weighted Avg**: Precision 0.72, Recall 0.71, F1-Score 0.70
- **Macro Avg**: Precision 0.74, Recall 0.62, F1-Score 0.65

Experiment 3: Support Vector Machines (SVM)

- **Accuracy**: 71.24%
- **Macro Avg**: Precision 0.69, Recall 0.66, F1-Score 0.67
- **Weighted Avg**: Precision 0.71, Recall 0.71, F1-Score 0.71

Experiment 4: Random Forest

- **Accuracy**: 56.76%
- **Macro Avg**: Precision 0.65, Recall 0.48, F1-Score 0.51
- **Weighted Avg**: Precision 0.63, Recall 0.57, F1-Score 0.57



| Model | Accuracy | Macro Avg Precision | Macro Avg Recall | Macro Avg F1-Score | Weighted Avg Precision | Weighted Avg Recall | Weighted Avg F1-Score |
|---|---|---|---|---|---|---|---|
| Logistic Regression | 73.11% | 0.73 | 0.68 | 0.70 | 0.73 | 0.73 | 0.73 |
| Multinomial Naive Bayes | 70.56% | 0.74 | 0.62 | 0.65 | 0.72 | 0.71 | 0.70 |
| Support Vector Machines | 71.24% | 0.69 | 0.66 | 0.67 | 0.71 | 0.71 | 0.71 |
| Random Forest | 56.76% | 0.65 | 0.48 | 0.51 | 0.63 | 0.57 | 0.57 |

**Table 6**

**Conclusion of Experiments**

- **Logistic Regression** achieved the highest accuracy of 73.11%, but the performance across categories varied significantly, especially in the "CULTURE & ARTS" category, where the precision was relatively low.
- **Multinomial Naive Bayes** showed a slightly lower accuracy of 70.56%, with notable strengths in the "FOOD & DRINK" and "PARENTING" categories, but struggled with categories like "COLLEGE" and "ENVIRONMENT."
- **Support Vector Machines (SVM)** yielded an accuracy of 71.24%, with consistent performance across most categories, although it underperformed in "ENVIRONMENT."
- **Random Forest** demonstrated the lowest accuracy of 56.76%, indicating challenges in categorizing news articles effectively, particularly in the "CULTURE & ARTS" and "ENVIRONMENT" categories.

The experiments highlight that while Logistic Regression provided the highest overall accuracy, the nuanced performance of Multinomial Naive Bayes and SVM in certain categories should not be overlooked. Future work may involve exploring advanced models or ensemble methods to enhance classification accuracy and robustness across all categories.

# CONCLUSIONS

This project successfully demonstrates an end-to-end news classification, summarization, and translation system, utilizing a comprehensive tech stack. On the backend, machine learning models were trained and optimized for accurate news categorization. Logistic Regression emerged as the best-performing model with the highest accuracy, while additional models like Multinomial Naive Bayes, SVM, and Random Forest were evaluated for performance comparisons. The backend also incorporates Flask to serve the trained models, while the integration of the News API allowed for real-time news fetching and prediction.

On the frontend, a user-friendly interface was developed using React.js. The platform allows users to view categorized news, summarized for quick reading, and translated into their preferred language. This not only enhances user experience but also ensures accessibility to a wider audience. The overall project highlights the synergy between advanced machine learning techniques and a seamless user interface, offering a tailored and efficient way for users to access relevant news content.

# REFERENCES

- News category classification video: https://youtu.be/6f4B4-KjbAw
- mysql connection with python: https://youtu.be/3vsC05rxZ8c
- flask setup: https://youtu.be/6M3LzGmIAso
- react falsk python integration: https://youtu.be/7LNl2JlZKHA
- newseverything api for daily news: https://newsapi.org/
- taken reference from this project for flow of the classification module: https://earthly.dev/blog/build-news-classifier-nlp-newsapi-lr/
- Research Paper:
  1. https://onlinelibrary.wiley.com/doi/full/10.1155/2023/1784394
  2. https://link.springer.com/chapter/10.1007/3-540-36127-8_20
  3. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=News+Aggregation+using+Web+Scraping+News+Portals&btnG=
  4. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Text+Classification+Using+Machine+Learning+Techniques&btnG=
  5. https://www.mdpi.com/2073-431X/12/1/16