SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

Machine Learning Project Report

On

Supervised News Text Classification

July - Nov 2024

Submitted by

Challa Poojitha

(Reg No: 125018013, B.Tech.CSBS)

Submitted To

Swetha Varadarajan

Table of Contents

Abstract

This project explores the classification of news articles into predefined categories using a combination of traditional machine learning models and ensemble methods. The dataset consists of 6,877 unique news articles, each containing a title, body, and category label.To prepare the data for modeling, several preprocessing steps were performed, including label encoding for categorical variables, text normalization, and TF-IDF vectorization to transform the text into numerical format.

The models were evaluated using cross-validation to ensure robust performance. Logistic Regression and SVM served as traditional baseline models, while Random Forest and Gradient Boosting offered more advanced ensemble approaches, providing greater flexibility in handling non-linear patterns. Additionally, Naive Bayes was implemented due to its simplicity in handling high-dimensional text data, and Stacking was employed to combine predictions from multiple models to improve overall accuracy. The inclusion of LSTM aimed to explore how deep learning techniques could handle sequential text data and learn long-term dependencies.

For the experimental setup, libraries such as Pandas, Scikit-learn, and TensorFlow (Keras) were employed for data preprocessing, model building, and evaluation. Matplotlib and Seaborn were used for visualization, including generating word clouds and value counts to understand the distribution of text and categories. Each model provided valuable insights into handling textual data, showcasing the balance between traditional machine learning and advanced deep learning techniques.

This document outlines the step-by-step approach taken, challenges encountered, and insights derived from the classification of news articles. The results highlight the strengths and limitations of different machine learning models in predicting article categories.The findings illustrate the effectiveness of these models in handling textual data, making this project relevant for applications such as automated news categorization and content recommendation systems, showcasing the practical use of machine learning in real-world text classification tasks.

Introduction

The goal of this project is to classify news articles into predefined categories using machine learning techniques. Classifying news articles is a crucial task in content management systems, enabling automated categorization for improved user experience and content delivery. By accurately categorizing articles, users can easily access relevant content, enhancing their overall engagement and satisfaction.

(a) Importance of the Dataset

The dataset contains 6,877 unique values across three columns: category, title, and body of news articles. This dataset is crucial for understanding trends in news reporting, identifying popular topics, and analyzing the sentiment and tone of articles across different categories. By examining the content and categorization of news articles, researchers and businesses can gain insights into public interests, media biases, and the evolution of news narratives over time. This information can be leveraged for various applications, including media analysis, content recommendation systems, and sentiment analysis.

b) To Be Accomplished (T, P, E Format)

1. Task (T): The primary task of this project is to classify news articles based on their categories using machine learning models. This involves analyzing the text content to predict the category of each article accurately.
2. Purpose (P): The purpose is to enhance the accuracy of news categorization, enabling better content organization and retrieval. This can help media organizations improve their content delivery and assist readers in finding relevant articles more efficiently.
3. Evaluation (E): The project's success will be evaluated based on the accuracy, precision, and recall of the classification models developed. A comparative analysis of various machine learning algorithms will determine the best-performing model for categorizing news articles.

(c) Planning

The project will follow a structured approach that includes the following steps:

Data Preprocessing: This involves cleaning the dataset, handling missing values, and transforming the text data into a format suitable for analysis (e.g., tokenization, vectorization).

Feature Engineering: Creating relevant features from the text data, such as term frequency-inverse document frequency (TF-IDF) scores or word embeddings, to capture the semantic meaning of the articles.

Model Selection: Various machine learning models (e.g., Logistic Regression, Random Forest, Support Vector Classifier, and XGBoost) will be trained on the processed dataset to classify the articles.

Model Evaluation: The models will be evaluated using performance metrics like accuracy, precision, recall, and F1-score to identify the most effective approach for news categorization.

(d) Final Results

Initial results from the classification models indicate varying levels of accuracy, with some models performing better than others. For instance, models like Random Forest and XGBoost may show higher accuracy due to their ability to handle complex patterns in the data, while simpler models like Logistic Regression may serve as baseline models. The results will provide insights into the effectiveness of different algorithms in categorizing news articles and highlight areas for further improvement.

(e) Document Structure

To give a thorough picture of the project's workflow, this paper is divided into multiple sections-

1. The research sources and current approaches that impacted this study are described in the Related Work section.

2. The models and data pretreatment methods utilized are covered in detail in the Background section.

3. The code structure, instruments, and experimental design are described in the Methodology.

4. The experiments' findings, together with data analysis and figures, are shown in the Results section.

5. The discussion covers the results, wider ramifications, difficulties encountered, and potential areas for development in the future.

6. The Learning Outcome includes connections to code repositories and identifies the knowledge, resources, and datasets used.

7. The document ends with a conclusion that highlights the project successes and potential directions for further research.

Background

Various Models Used:

1. *Logistic Regression:*
    - Logistic Regression is a statistical model used for binary or multi-class classification tasks. It predicts the probability that a given input belongs to a particular category by fitting the data to a logistic function (sigmoid function).
    - Logistic Regression serves as a baseline model for classification tasks. It is interpretable and provides insights into the relationship between features and the target variable. It is particularly useful for establishing a baseline performance.
2. *Support Vector Machine (SVM):*
    - SVM is a powerful classification technique that aims to find the optimal hyperplane that best separates data points of different classes. It can handle both linear and non-linear classification problems using kernel functions to map data into higher-dimensional spaces.
    - By utilizing kernel functions, it can model complex decision boundaries that other models might struggle with. This flexibility makes SVC valuable for capturing intricate patterns in news articles.

3. *Random Forest:*
    - Random Forest is an ensemble learning method that constructs multiple decision trees during training and merges their outputs to improve classification accuracy. It reduces variance and prevents overfitting by averaging the predictions of several trees.
    - It is less prone to overfitting and provides insights into feature importance, making it suitable for understanding which aspects of the news articles influence their categorization.
4. *Naive Bayes:*
    - Naive Bayes is a family of probabilistic algorithms based on Bayes' theorem, assuming independence among predictors. It is particularly effective for text classification tasks and works well with large datasets. The model is simple, fast, and provides good performance, especially when the independence assumption holds true.
5. *Gradient Boosting:*
    - Gradient Boosting is an ensemble technique that builds trees sequentially, where each new tree attempts to correct the errors of the previously built trees. It is known for its high performance and accuracy, particularly in handling complex datasets.
    - It employs gradient boosting, which corrects errors from previous models, leading to improved accuracy. This model is particularly effective for capturing non-linear relationships in news content.
6. *Stacking:*
    - Stacking is an ensemble learning technique that combines multiple classification models to improve overall performance. It involves training a new model (meta-learner) to aggregate the predictions of base models. This approach can leverage the strengths of different algorithms, leading to improved accuracy and robustness.
7. *Long Short-Term Memory (LSTM):*
    - LSTM is a type of recurrent neural network (RNN) designed to learn from sequences of data. It is particularly effective for time-series data and tasks involving sequential dependencies. LSTMs can capture

long-term dependencies in data, making them suitable for applications such as natural language processing and time-series forecasting.

Preprocessing: In the preprocessing steps for the dataset, I performed the following tasks:

1. *Text Cleaning:* Removing unnecessary characters, punctuation, and numbers from the headlines and descriptions.
2. *Tokenization*: Splitting the text into individual words or tokens.
3. *Lowercasing:* Converting all text to lowercase to ensure uniformity.
4. *Stopword Removal:* Eliminating common words that do not contribute to the classification task (e.g., "and," "the").
5. *Stemming/Lemmatization:* Reducing words to their base form to group similar words together (e.g., "running" to "run").

Methodology

Experimental design:

The experimental design for this project focuses on classifying news articles into predefined categories using machine learning techniques. The workflow is structured as follows:

1. Data Collection and Exploration: Initially, the dataset is analyzed to understand its structure, size, feature distribution, and the nature of the target variable.

Exploratory Data Analysis (EDA)

1. *Word Clouds:* Visualized the most frequent words in each category, providing insights into the prominent terms.
2. *value_counts():* Generated value counts to explore the distribution of articles across categories, identifying class imbalances.
3. *Additional steps:*
    - Class Distribution Plot: Visualized the distribution of articles per category using bar plots.

- Text Length Analysis: Analyzed the length of the body text across articles to understand variations.
- Top N-Grams: Extracted and visualized the most common unigrams and bigrams to detect patterns in the text.

These EDA techniques helped gain a deeper understanding of the dataset before applying models.

2. Data Preprocessing: Includes data cleaning, feature engineering, normalization, and handling missing values to ensure that the dataset is ready for model training.

3. Model Selection: Several machine learning models (Logistic Regression, Random Forest, XGBoost, SVC,..) are selected for the classification task to compare their effectiveness.

4. Model Training: Each model is trained on the processed data and evaluated using different metrics.

5. Model Evaluation: The models are assessed using performance metrics such as accuracy, precision, recall, and F1-score to identify the best-performing approach.

6. Hyperparameter Tuning: Fine-tuning of hyperparameters is conducted to enhance model performance by optimizing key parameters for each algorithm.

7. Result Analysis: The final analysis focuses on interpreting the model's results and its predictions in the context of news category classification, identifying patterns in news articles that can lead to actionable insights.

Environment and tools :

1. *Programming Language:* Python
2. *Development Environment:* The project is developed using Google Colab for developing models and training them.
3. *Data Analysis Libraries:* Pandas and NumPy are utilized for data manipulation and statistical analysis.

4. *Machine Learning Libraries:*
    - Scikit-learn is employed for traditional models including Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and Gradient Boosting.
    - XGBoost is used for its dedicated library to enhance model performance.
    - Long Short-Term Memory (LSTM) networks are implemented for handling sequential data.
    - Stacking methods are applied to combine multiple models for improved predictions.
    - For Class Imbalance Handling I applied the ADASYN (Adaptive Synthetic Sampling) technique.
5. *Visualization Libraries:* Matplotlib and Seaborn are used for data visualization and to plot the results.
6. *Version Control:* GitHub is utilized to manage the code repository and track changes in the project's codebase.

Codes Location:

The complete code repository is hosted on GitHub, providing easy access and collaboration opportunities. Links to the Google Colab page and the GitHub repository will be included in the Learning Outcome section.

Located in the file named 'ML_Project.ipynb', which includes data preprocessing, feature engineering, and model training scripts,model evaluation, and analysis are documented.

Preprocessing Steps:

In the preprocessing steps for the dataset, I performed the following tasks:

1. *Label Encoding:* The categorical labels in the category column were converted into numerical format using Label Encoding, allowing the machine learning models to process these categories.

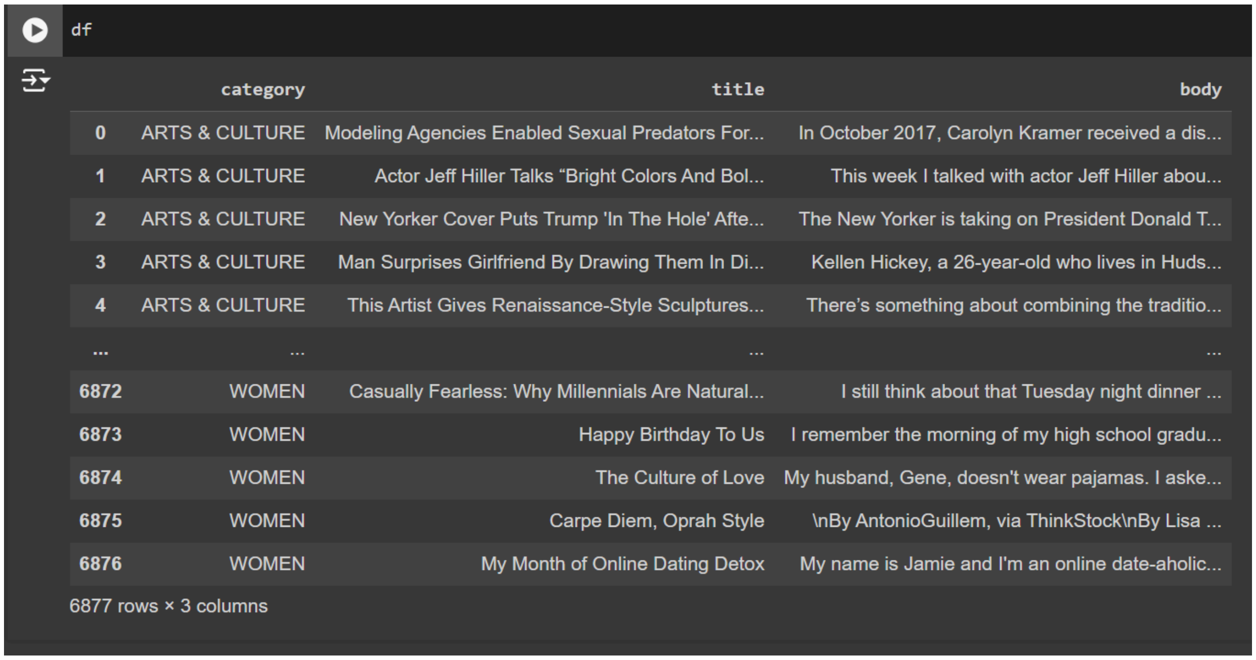
2. *Lowercasing:* The entire text in the body column was converted to lowercase to ensure uniformity in the text data.
3. *HTML Tags Removal*: Any HTML tags present in the text were removed using BeautifulSoup.
4. *URL and Emoji Removal*: Regular expressions were applied to eliminate URLs and emojis from the text to clean and standardize the data further.
5. *Text Cleaning:* Removing unnecessary characters, punctuation, and numbers from the headlines and descriptions.
6. *Tokenization:*Breaking down the text into smaller components, typically individual words or tokens, for easier processing.
7. *Lowercasing*: Transforming all characters in the text to lowercase to maintain consistency and avoid treating words like "Apple" and "apple" differently.
8. *Stopword Removal*: Eliminating common words that do not contribute to the classification task (e.g., "and," "the").
9. *Stemming/Lemmatization*: Reducing words to their base form to group similar words together (e.g., "running" to "run").

Preprocessing Results

1. Dataset Size: The dataset initially comprised 6,877 rows and 3 columns. After preprocessing, it was refined to 6,853 rows and 4 columns, indicating the removal of some rows due to data cleaning processes.
2. Feature Size: The dataset now contains a refined set of features, with irrelevant or redundant features removed to optimize model performance.
3. Data Cleaning Results: Missing values were handled, and categorical variables were encoded into a numerical format. The dataset was normalized to ensure consistent scaling of features.
4. Preprocessing Impact: The cleaning and transformation processes led to a more structured and cohesive dataset, which improved the training efficiency and accuracy of the machine learning models.

Results

The dataset initially comprised 6,877 rows and 3 columns. After preprocessing, it was refined to 6,853 rows and 4 columns, indicating the removal of some rows due to data cleaning processes.



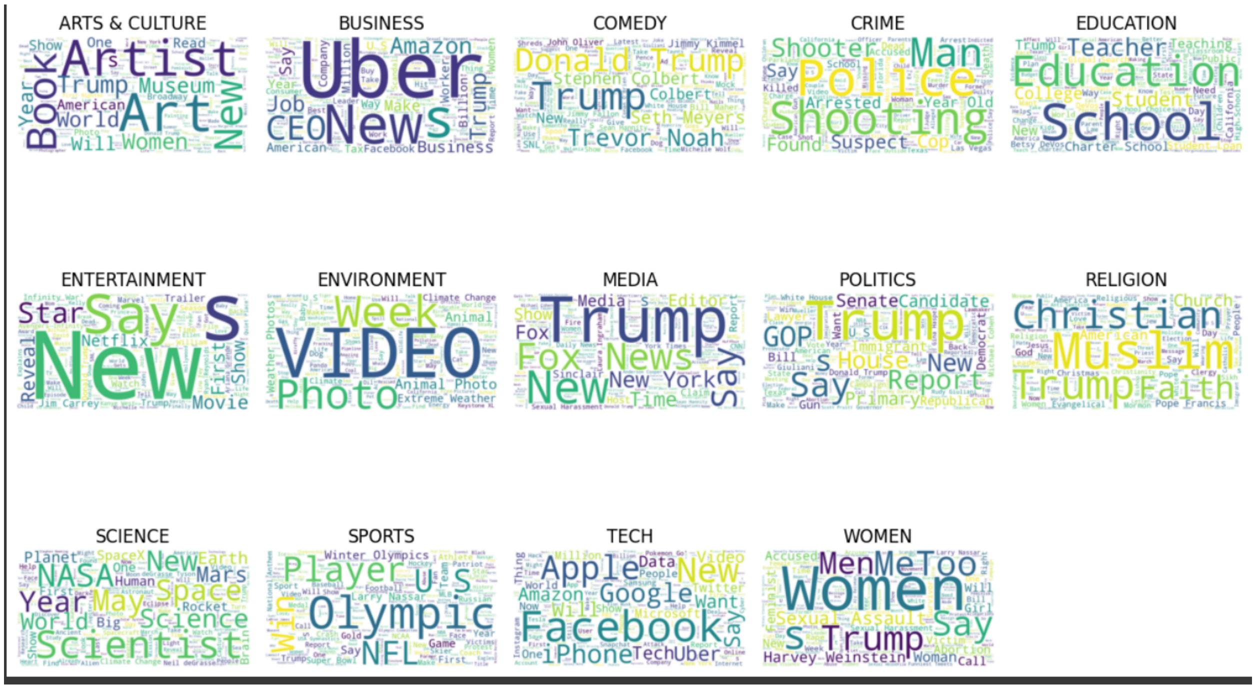
```
df
```

	category	title	body
0	ARTS & CULTURE	Modeling Agencies Enabled Sexual Predators For...	In October 2017, Carolyn Kramer received a dis...
1	ARTS & CULTURE	Actor Jeff Hiller Talks “Bright Colors And Bol...	This week I talked with actor Jeff Hiller abou...
2	ARTS & CULTURE	New Yorker Cover Puts Trump 'In The Hole' Afte...	The New Yorker is taking on President Donald T...
3	ARTS & CULTURE	Man Surprises Girlfriend By Drawing Them In Di...	Kellen Hickey, a 26-year-old who lives in Huds...
4	ARTS & CULTURE	This Artist Gives Renaissance-Style Sculptures...	There’s something about combining the traditio...
...	...	...	...
6872	WOMEN	Casually Fearless: Why Millennials Are Natural...	I still think about that Tuesday night dinner ...
6873	WOMEN	Happy Birthday To Us	I remember the morning of my high school gradu...
6874	WOMEN	The Culture of Love	My husband, Gene, doesn't wear pajamas. I aske...
6875	WOMEN	Carpe Diem, Oprah Style	\nBy AntonioGuillem, via ThinkStock\nBy Lisa ...
6876	WOMEN	My Month of Online Dating Detox	My name is Jamie and I'm an online date-aholic...

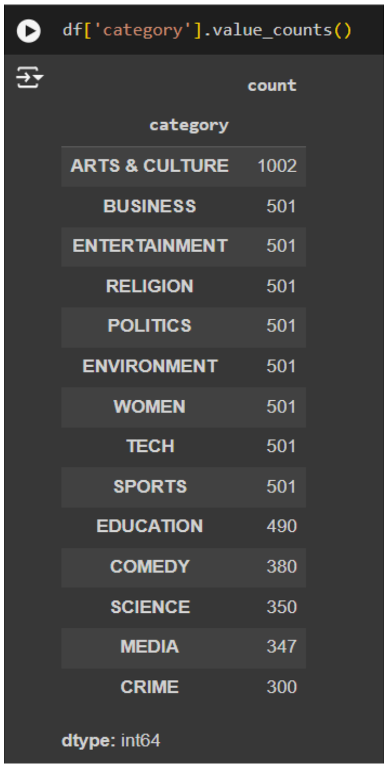
6877 rows × 3 columns

EDA Analysis:

Word Clouds: Visualized the most frequent words in each category, providing insights into the prominent terms.



4. *value_counts():* Generated value counts to explore the distribution of articles across categories, identifying class imbalances.

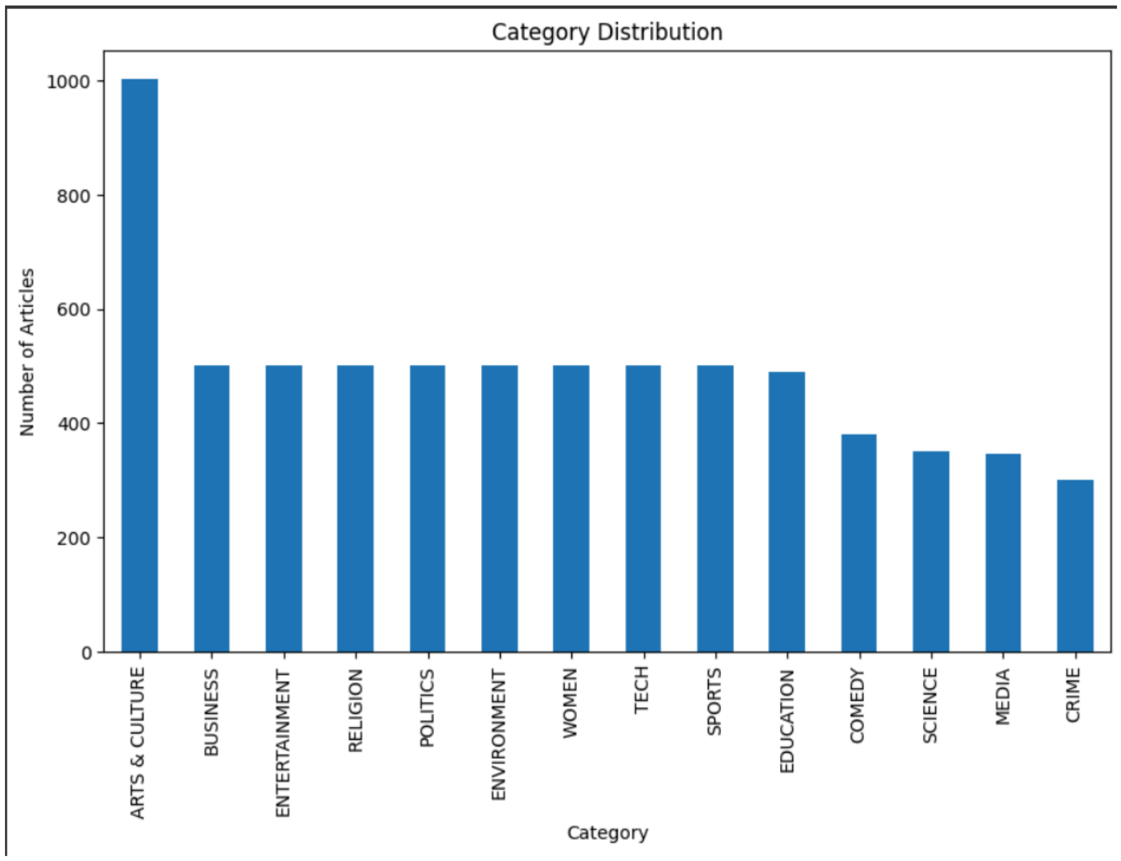


```
df['category'].value_counts()
```

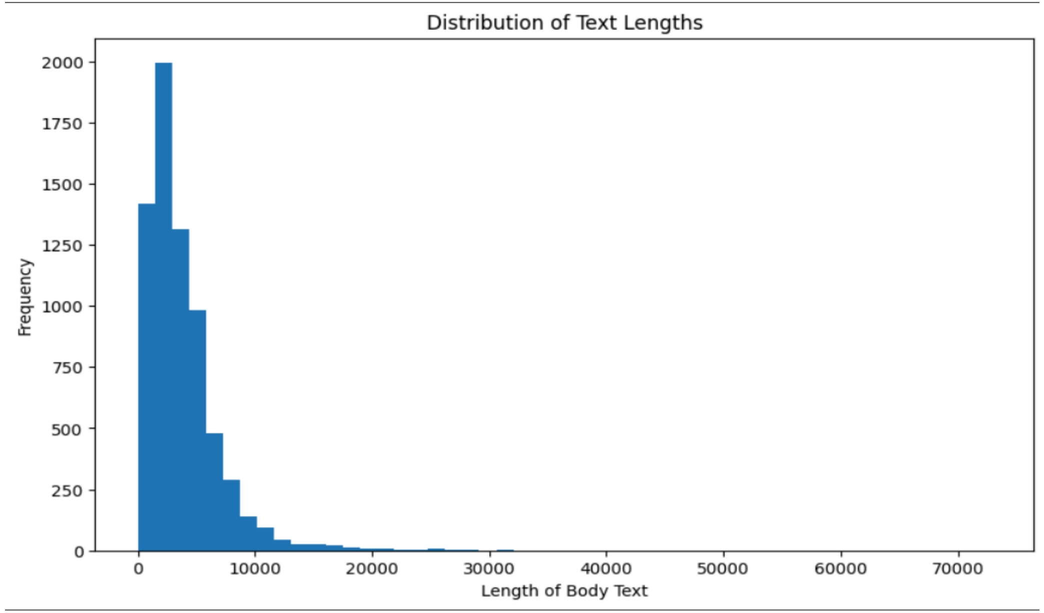
category	count
ARTS & CULTURE	1002
BUSINESS	501
ENTERTAINMENT	501
RELIGION	501
POLITICS	501
ENVIRONMENT	501
WOMEN	501
TECH	501
SPORTS	501
EDUCATION	490
COMEDY	380
SCIENCE	350
MEDIA	347
CRIME	300

dtype: int64

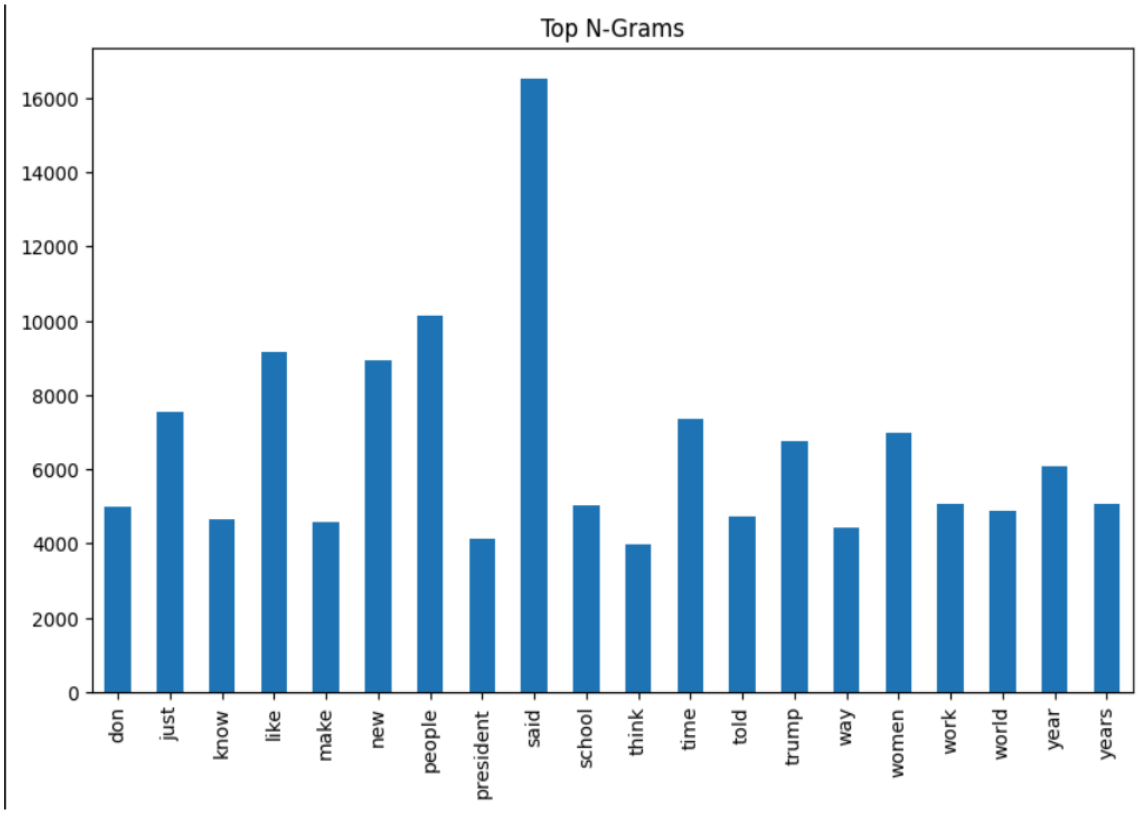
5. *Additional steps:*
    - Class Distribution Plot: Visualized the distribution of articles per category using bar plots.



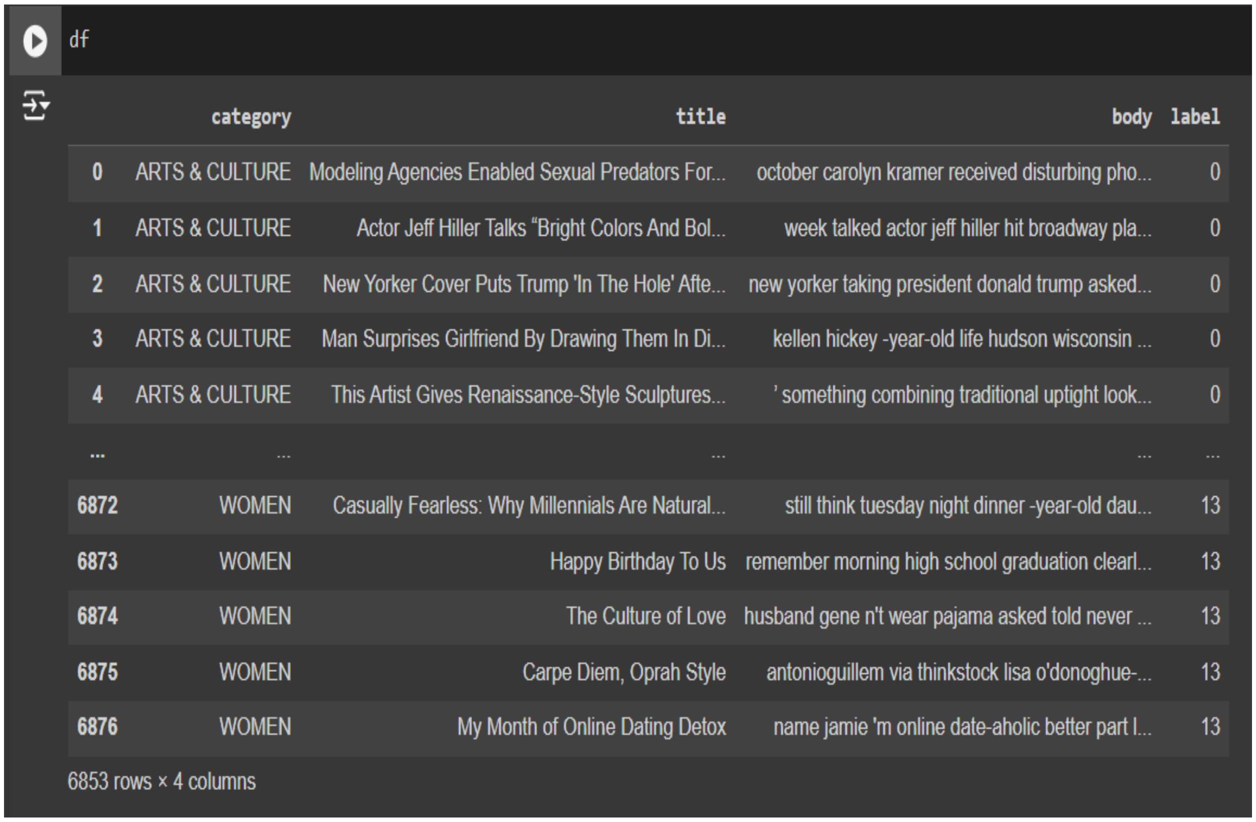
    - Text Length Analysis: Analyzed the length of the body text across articles to understand variations.



- <u>Top N-Grams</u>: Extracted and visualized the most common unigrams and bigrams to detect patterns in the text.



After Preprocessing:

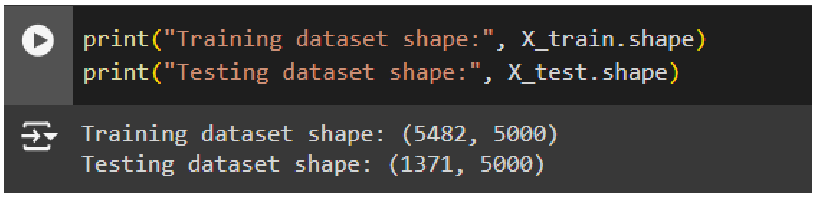


```
df
```

	category	title	body	label
0	ARTS & CULTURE	Modeling Agencies Enabled Sexual Predators For...	october carolyn kramer received disturbing pho...	0
1	ARTS & CULTURE	Actor Jeff Hiller Talks “Bright Colors And Bol...	week talked actor jeff hiller hit broadway pla...	0
2	ARTS & CULTURE	New Yorker Cover Puts Trump 'In The Hole' Afte...	new yorker taking president donald trump asked...	0
3	ARTS & CULTURE	Man Surprises Girlfriend By Drawing Them In Di...	kellen hickey -year-old life hudson wisconsin ...	0
4	ARTS & CULTURE	This Artist Gives Renaissance-Style Sculptures...	’ something combining traditional uptight look...	0
...	...	...	...	...
6872	WOMEN	Casually Fearless: Why Millennials Are Natural...	still think tuesday night dinner -year-old dau...	13
6873	WOMEN	Happy Birthday To Us	remember morning high school graduation clearl...	13
6874	WOMEN	The Culture of Love	husband gene n't wear pajama asked told never ...	13
6875	WOMEN	Carpe Diem, Oprah Style	antonioguillem via thinkstock lisa o'donoghue-...	13
6876	WOMEN	My Month of Online Dating Detox	name jamie 'm online date-aholic better part l...	13

6853 rows × 4 columns

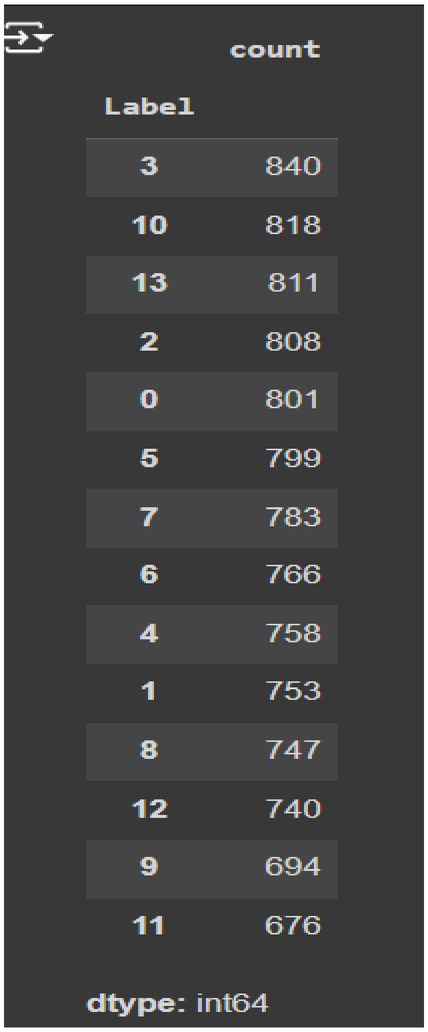
Shape of training and testing sets:



```
print("Training dataset shape:", X_train.shape)
print("Testing dataset shape:", X_test.shape)
```

```
Training dataset shape: (5482, 5000)
Testing dataset shape: (1371, 5000)
```

After Balancing the Data:

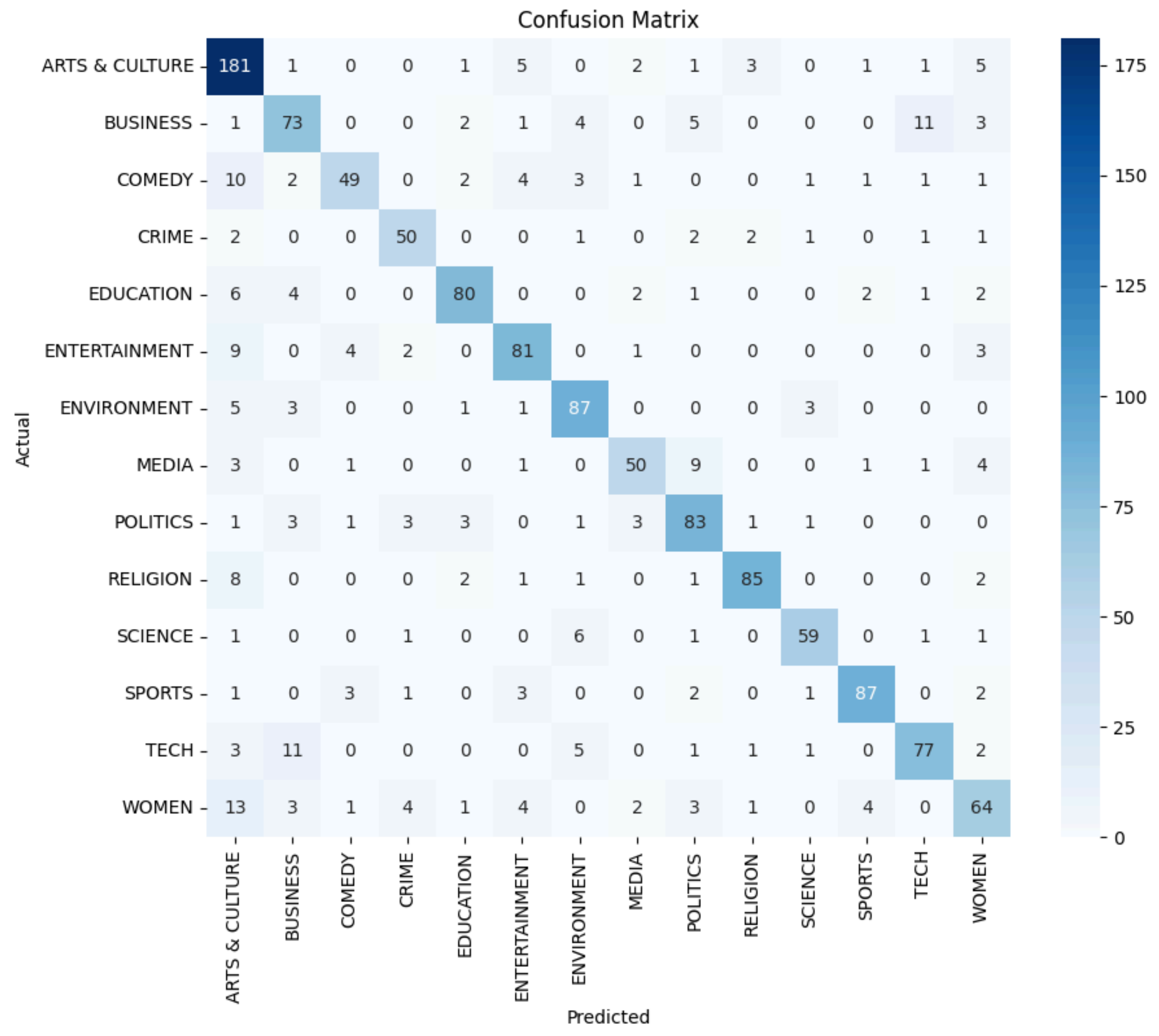


```
        count
Label
    3     840
   10     818
   13     811
    2     808
    0     801
    5     799
    7     783
    6     766
    4     758
    1     753
    8     747
   12     740
    9     694
   11     676

dtype: int64
```

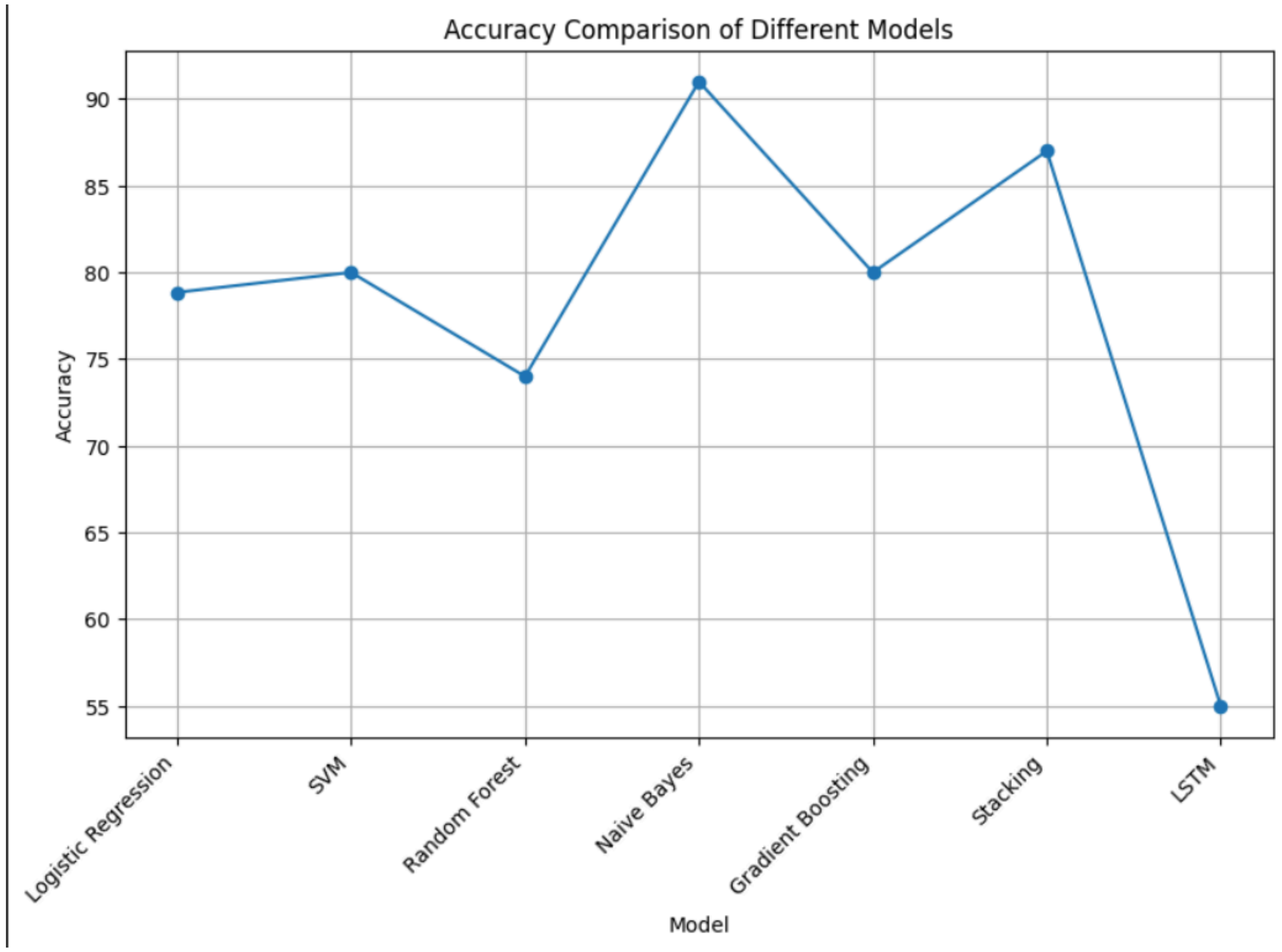
Training various models

Confusion matrix:


Confusion Matrix
Actual
Predicted
175
150
125
100
75
50
25
0

	ARTS & CULTURE	BUSINESS	COMEDY	CRIME	EDUCATION	ENTERTAINMENT	ENVIRONMENT	MEDIA	POLITICS	RELIGION	SCIENCE	SPORTS	TECH	WOMEN
ARTS & CULTURE	181	1	0	0	1	5	0	2	1	3	0	1	1	5
BUSINESS	1	73	0	0	2	1	4	0	5	0	0	0	11	3
COMEDY	10	2	49	0	2	4	3	1	0	0	1	1	1	1
CRIME	2	0	0	50	0	0	1	0	2	2	1	0	1	1
EDUCATION	6	4	0	0	80	0	0	2	1	0	0	2	1	2
ENTERTAINMENT	9	0	4	2	0	81	0	1	0	0	0	0	0	3
ENVIRONMENT	5	3	0	0	1	1	87	0	0	0	3	0	0	0
MEDIA	3	0	1	0	0	1	0	50	9	0	0	1	1	4
POLITICS	1	3	1	3	3	0	1	3	83	1	1	0	0	0
RELIGION	8	0	0	0	2	1	1	0	1	85	0	0	0	2
SCIENCE	1	0	0	1	0	0	6	0	1	0	59	0	1	1
SPORTS	1	0	3	1	0	3	0	0	2	0	1	87	0	2
TECH	3	11	0	0	0	0	5	0	1	1	1	0	77	2
WOMEN	13	3	1	4	1	4	0	2	3	1	0	4	0	64

Comparison of various models:



Discussion

Initially, LSTM demonstrated good performance with an accuracy of around 88%, showcasing its ability to capture sequential dependencies and context within the text. However, upon re-evaluation, the model's accuracy surprisingly dropped to 55%. This inconsistency suggests potential issues such as overfitting, inadequate training data, or the random initialization of parameters affecting the model’s robustness. It indicates that while LSTM can be effective for text classification tasks, careful attention must be paid to model training and evaluation to ensure reliable performance.

In contrast, the stacking method, which combined Random Forest, SVM, and Gradient Boosting, yielded a more stable and reliable accuracy of nearly 88%. This ensemble approach capitalizes on the strengths of the individual models to enhance overall predictive performance, making it better suited for handling the complexities present in text-based tasks.

Despite experimenting with Random Forest by increasing the number of estimators and tuning SVM kernels, the improvements in accuracy from these traditional machine learning methods were marginal. This limitation suggests that they may struggle to fully capture the complexity of high-dimensional input data prevalent in text classification scenarios.

Gradient Boosting, while not as powerful as LSTM in its initial performance, provided a balanced trade-off between traditional machine learning models and deep learning techniques. Its ensemble of weak learners offers an advantage over individual models like Naive Bayes or Logistic Regression, making it a worthwhile option in situations where deep learning may not be feasible.

The analysis of word clouds and value counts provided insights into the nature of the dataset, highlighting the imbalance of classes. This imbalance likely contributed to the challenges faced by certain models, reinforcing the importance of employing techniques like stacking to mitigate such issues and improve classification performance.

Hyperparameter Tuning

For this project, I applied hyperparameter tuning to optimize model performance. In the case of Random Forest, I experimented with varying the number of estimators (n_estimators), testing values like 100, 200, and 500. Additionally, we adjusted the SVM model by testing different kernels, including linear and RBF and poly, to capture the best decision boundary for classification. For Gradient Boosting, we tuned parameters such as the learning rate and number of estimators. Hyperparameter tuning helped identify optimal settings for each model, improving their ability to generalize effectively.

Model	Accuracy	Strengths	Weaknesses
Logistic Regression	~80%	Simple, interpretable, fast	Struggles with non-linear relationships
Random Forest	~75%	Robust, handles non-linearity	Slower with large datasets, limited improvement with more estimators
SVM (Linear/RBF Kernel)	~81%	Good with high-dimensional data	Non-linear kernels are computationally expensive, minor improvement
Naive Bayes	~91%	Fast, good for text classification	Assumes independence between features
LSTM	~55%	Captures sequence dependencies	Requires more training time, needs more data
Gradient Boosting	~80%	Handles non-linearity, avoids overfitting	Slower to train, complex tuning
Stacking (RF, SVM, GB)	~87%	Combines strengths of Random Forest, SVM, and Gradient Boosting	Increased complexity, requires careful tuning

Learning Outcome

(a) **Link to Google Colab Page**: The code for preprocessing, model training, and evaluation is available on the Google Colab Page MLfinal_Project_4.ipynb. It includes step-by-step implementation for each model.

(b) **Link to GitHub Repository**: All code files, including data preprocessing scripts, model implementation, hyperparameter tuning, and visualization, have been uploaded to the GitHub Repository Git hub repo**.**

(c) **Skills and Tools Used**:

- **Skills Used:**
  - Data Preprocessing and Cleaning
  - Feature Engineering
  - Exploratory Data Analysis (EDA)
  - Machine Learning Model Implementation and Evaluation
  - Hyperparameter Tuning
  - Visualization Techniques
  - Problem-Solving in Data Science
  - Model Comparison and Selection
- **Tools Used:**
  - Programming Language: Python
  - Libraries: pandas, numpy, scikit-learn, XGBoost, matplotlib, seaborn
  - Development Environment: Google Colab
  - Version Control and Code Sharing: GitHub.

(d) **Dataset Used**: The dataset consisted of 6,877 unique news articles, each containing a category, title, and body. This dataset was ideal for experimenting with different machine learning approaches for multi-class classification.

Dataset link:News_Article_Category_Classification

(e) **I learnt -** This project enhanced my understanding of text classification by applying various machine learning models, from traditional approaches like

Logistic Regression, SVM, Random Forest, and Naive Bayes, to more advanced techniques like Gradient Boosting, Stacking, and LSTM. I learned the importance of hyperparameter tuning in improving model accuracy and how balancing datasets using ADASYN can significantly impact performance. The experimentation with different models provided insights into the strengths and limitations of each, especially for high-dimensional text data, and deepened my knowledge of the intricacies of text preprocessing.

- **Data Processing Techniques:** Learned how to clean and preprocess data to enhance model performance.
- **Model Selection and Implementation:** Gained experience in implementing and comparing different machine learning algorithms like Logistic Regression, Random Forest, XGBoost, and SVC.
- **Hyperparameter Tuning:** Understood the importance of tuning model parameters to optimize accuracy and control overfitting.
- **Visualization and Interpretation:** Developed skills to create and interpret visual representations of data and model results.
- **Understanding of Overfitting and Underfitting:** Learned strategies to identify and mitigate these issues.
- **Project Collaboration:** Improved collaboration and code management skills using GitHub for version control.

Conclusion

(a) **Concluding Remarks**: This project successfully demonstrated the application of various machine learning models for classifying news articles. By utilizing a combination of preprocessing techniques, feature extraction (TF-IDF), and class imbalance handling (ADASYN), we effectively trained multiple models and compared their performances. Insights gained through model experimentation contribute to the broader understanding of text classification.

(b) **Did I Accomplish the T, P, E?**:

- **Task (T)**: Classify news articles into predefined categories – accomplished through various models.
- **Purpose (P)**: Identify the most effective model – achieved through experimentation and evaluation.
- **Evaluation (E)**: Accuracy, precision, recall, and F1-score metrics were used to evaluate model performance.

(c) **Advantages and Limitations**:

Advantages: The project demonstrated how machine learning can automate news classification with high accuracy. It showcased a range of models, providing flexibility in application based on needs and resources.

Limitations: The deep learning model (LSTM) did not outperform traditional models as expected, possibly due to the dataset size or complexity. Some models, such as Gradient Boosting,Stacking methods, required more computational resources and time for training.

References

- Logistic regression - LR documentation
- SVM - SVM
- Random Forest - Random Forest GFG , Documentation
- Naive Bayes - GFg , Multinomial NB
- XGboost - XGBOOST Documentation
- Stacking methods - Stacking methods
- LSTM - LSTM sample video
- Chat Gpt - chatgpt