Text Classification- Evaluating Three Classification Algorithms Across Two Distinct Datasets: A Comparative Analysis

Denniston, C.

Abstract

The primary objective was to enhance the performance of three algorithms, Multinomial Naive Bayes, MLP and SVM and improve data quality by employing a wide range of preprocessing techniques. In the first experiment, I achieved the best scores with raw datasets by optimising models with various parameters and using feature engineering techniques. In the second experiment, I continued to employ the optimised algorithms, testing them with processed datasets using different preprocessing techniques. For the 'Fake and Real News' dataset, the preprocessing techniques used in the experiment included stop word removal, SpaCy lemmatisation, as well as CountVectorizer and TF-IDF for feature engineering that were compared with Embedding, and tuning the algorithms to find best parameters. Through iterative experiments with these techniques, I identified a model, preprocessing method, and feature engineering technique that produced the best outcome, achieving a 0.99 for the three evaluations metrics (Precision, Recall and F1-score). The report also compares the similarities and differences between the three algorithms, preprocessing techniques, and feature engineering techniques in detail which provides valuable insights for future projects in the realm of text classification and sentiment analysis.

1- Introduction

Text classification, a key area in natural language processing (NLP), has gained significance with the advent of digital documents. It involves categorising text into topics or genres, covering various content types like scientific articles, emails, news reports, movie reviews, and ads. Classification is based on factors like creation, editing, language, and target audience. Data for text classification is primarily collected from the internet, spanning sources like newsgroups, bulletin boards, broadcasts, and social media. This data varies in format, vocabulary, and writing styles, even within the same category, and often falls into multiple categories simultaneously (Ikonomakis, Kotsiantis & Tampakas, 2005).

For an extended period, the primary emphasis was on numerical classification. However, over the past decade, there has been a notable shift in focus towards text classification. Both numerical and text classification share the common objective of categorising data points into specific labels, employing standard evaluation metrics, and relying on supervised machine learning algorithms. Nevertheless, they diverge in terms of data representation, feature engineering, model selection, and interpretability, mainly due to the distinct attributes of text data. Consequently, these distinctions call for distinct approaches and techniques (Macskassy et al., 2001).

This report presents the results of an experiment conducted on the distinct dataset, centered around the classification of free textual data. The dataset involves binary categorisation into "Fake and Real News”. The primary objective of this experiment is to analyse the results and findings by applying three distinct model algorithms: Multinomial Naïve Bayes (MNB), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM), to both unprocessed and preprocessed text data. Additionally, two different feature selection methods, namely vectoriser (CountVectorizer and Term Frequency-Inverse Document Frequency (TF-IDF)) and Embedded and Padding, have been implemented for the dataset.

The three models, Multinomial Naive Bayes (MNB), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM), are all commonly used for classification tasks and share the common requirement of labelled training data. However, they differ in several key ways. MNB operates on term frequency, counting word occurrences, and is rooted in probabilistic principles (Singh et al., 2019). In contrast, MLP is a deep neural network with multiple hidden layers, well-suited for tackling complex and non-linear problems. SVM, on the other hand, works in a vector space, with its primary goal being to find a decision surface (hyperplane) that effectively separates data into two distinct classes (Zhang, Yoshida & Tang, 2008).

The experiment is structured into several key stages, starting with methodology, followed by data description, experiments, results and discussion, and concluding with reflections and a conclusion. In the methodology phase, the research design, algorithms, data preprocessing, and feature selection methods are outlined. The data description section provides insights into the datasets, including their sources and characteristics. The experiment's stage details the setup, model performance metrics, preprocessing steps, and parameter tuning. Results and discussion present the findings, analyse them, and draw implications, while the conclusion and reflections summarise the experiment's contributions, limitations, and potential for future research.

2- Methodology

The experimental methodology applied to the Fake and Real News datasets is visually depicted in Figure 1.

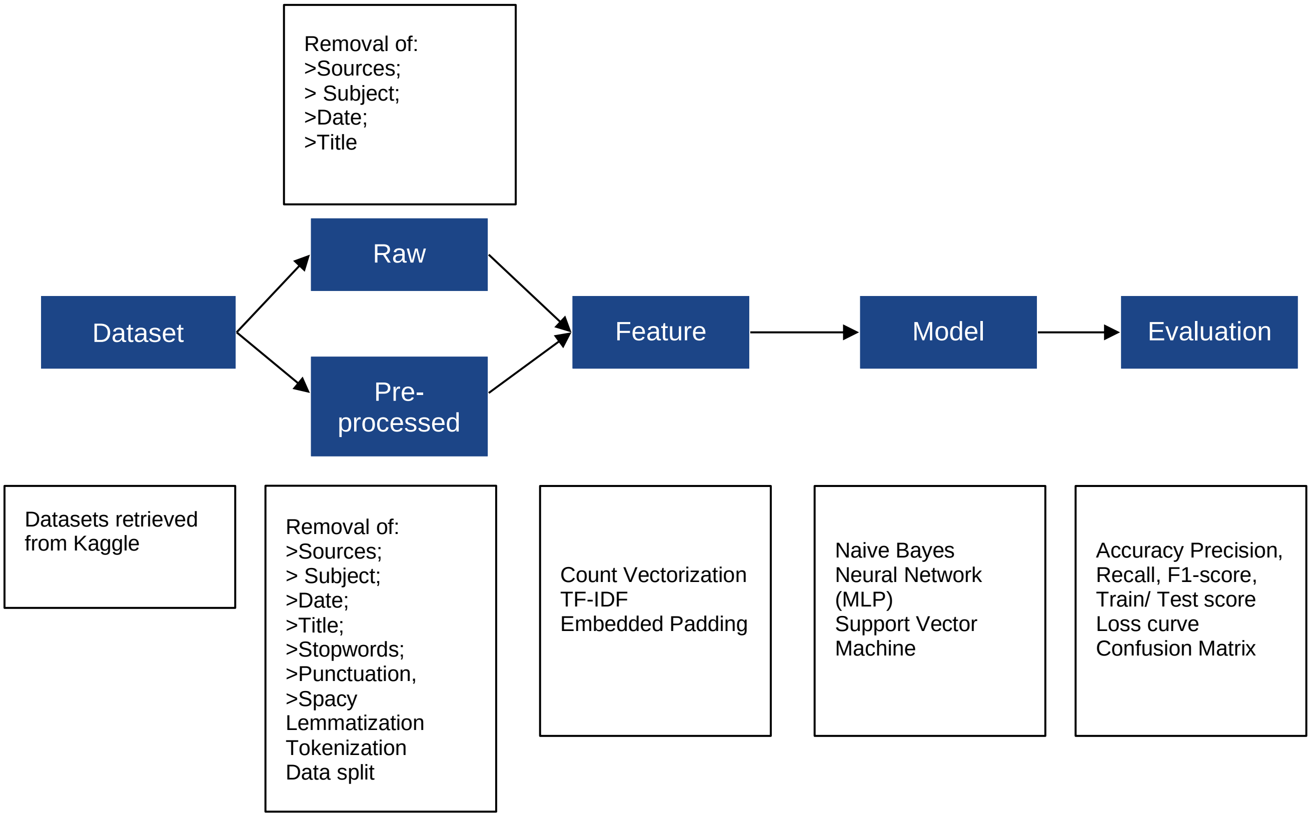


Figure 1: Experiment flowchart of learning algorithm for unprocessed and pre-processed datasets.

The dataset "Fake and Real News" was obtained from Kaggle. The initial phase involved exploratory data analysis (EDA), where I examined the datasets' characteristics, such as the number of features, target labels, and performed data cleaning tasks. This included removing non-textual columns, handling data unbalanced, duplicates, and addressing null values.
The experiment was divided into two parts:

1. Unprocessed Data:
I began by splitting the dataset into features (X) and target/ class (y). To transform textual data into numerical vectors, I applied two techniques: Count Vectorization and TF-IDF Transformer. Count Vectorization counted word frequencies, while TF-IDF calculated values that considered word importance and document length. It introduced a weighting system by combining Term Frequency (TF) and Inverse Document Frequency (IDF), resulting in a balanced representation, particularly beneficial for common words (Wendland, Zenere, & Niemann, 2021).

The approach was expended by incorporating pre-trained word embeddings, specifically GloVe word vectors. This involved label encoding, tokenisation, and padding. I also constructed an embedding matrix and assessed its coverage. Subsequently, split the vectorised data into an 80% stratified sample for model training.

I developed three models: Multinomial Naive Bayes (MNB), Multilayer Perceptron (MLP), and Support Vector Machine (SVM). Closely monitored their training progress and adjusted parameters to achieve the highest metrics, including Accuracy, Precision, Recall, and F1-score.

2. Preprocessed Data:
In the second part of the experiment, I aimed to enhance the model further. Text preprocessing included the removal of stopwords (common words like "the" and "and") to improve text analysis efficiency, Spacy Lemmatization (reducing words to their base form, e.g., "running" to "run") for better semantic analysis, and tokenisation (splitting text into individual words or tokens) to prepare it for further tasks.

The results were based on metrics such as Accuracy, Precision, Recall, and F1-score. I iteratively modified the preprocessing methods and reevaluated the models until it was achieved the highest performance. The outcomes from both unprocessed and preprocessed data were recorded, compared, and analysed to draw conclusions and provide recommendations.

3- Data Description

3.1 - Dataset: Fake and Real news dataset

Description: This dataset comprises two distinct categories of news articles: “Fake” and “True/Real” news. It consists of two separate CSV files. The “Fake.csv” file has a total of 23,481 entries with four columns, and the “True.csv” file contains 21,417 entries, also with four columns. These datasets were meticulously collected from a variety of reliable sources, including genuine articles, reputable news websites, and sources that have been scrutinised and labelled by Politifact, a prominent fact-checking organisation in the United States, as well as Wikipedia.
The dataset encompasses a wide range of article types and topics, with a notable concentration on political and global news subjects. Each entry in the dataset contains information on the article’s title, full text, its subject or category (such as Politics, World Politics, and more), and the date on which the article was published.
Data Fields:

1. Title: Title of the article;
2. Text: the article text;
3. Subject: type of article (Politics, World Politics, and so on);
4. Date: date the article was published.

Data Source: The dataset was generated from real world sources and different articles.
Data Quality: Dataset presented duplicates, free from missing values.

4- Experiments

Endeavoured to further improve the results from the raw dataset by preprocessing the data to address noise and enhance features. It was conducted experiments involving various combinations of preprocessing tasks to test the three algorithms. The following are preprocessing methods used, including but not limited to stop word removal, stemming and lemmatisation, feature size adjustment, and the type of vectorisation.

Identifying Optimal Preprocessing Techniques for Improved Results

During the exploratory data analysis (EDA) of the Fake and Real News dataset, initially was encountered two distinct dataset files: one containing fake news, referred to as 'fake_df,' and the other containing real news, labelled 'true_df.' These datasets consisted of 23,480 and 21,417 entries, respectively, and were found to be devoid of any null values. As a next step, the datasets were concatenated, creating a unified dataset in which fake news was represented as '0,' and true news as '1' in a new target column named 'class.' This allowed for a binary classification approach.

In the cleaning phase, certain columns, including 'date,' 'subject,' and 'title,' which were deemed unnecessary for analysis, were removed from the dataset. The retained columns were 'text' and 'class.' Additionally, during this process, identified and removed 6,252 duplicate entries from the combined dataset, resulting in a refined dataset.

Furthermore, it was analyzed the class distribution to check the balance between the target classes. It was revealed that true news comprised approximately 54.83% of the dataset, while fake news made up the remaining 45.17%, refer to appendix A.1. This distribution displayed a 9% difference, which was just below the 10% threshold for considering the dataset balanced. Subsequently, in EDA, it was explored the subject matter and found that the top three highest-frequency subjects in the dataset were "politicsNews," followed by "worldnews," and "news" (Appendix A.1).

Moving on to the text pre-processing, stop words were removed (e.g., "a," "and," "the") as they added minimal information to the target class or semantics. It was also applied contractions to replace common contractions with their expanded forms (e.g., "I'm" to "I am"). Lowercasing was employed to standardise the text data by converting all text to lowercase. URLs or web links were removed, and non-alphanumeric characters (i.e., special characters and punctuation) were stripped from the text. Tokenisation was used to split the text into individual words (tokens). Finally, lemmatisation was performed, which involved taking the list of tokens, joining them into a string, and applying lemmatisation using spaCy's natural language processing capabilities. This reduced words to their base or dictionary form. Analysis using the Counter (corpus) revealed that the top three high-frequency words were "say" (152,142), "trump" (110,245), and "president" (46,208). To prepare both unprocessed and preprocessed text data for feature selection, it was applied CountVectorizer and TF-IDF techniques. Additionally, Embedding and padding processes were carried out to enable the comparison of various machine learning models' performance. Subsequently, the data was split into an 80/20 train/test set using stratification, facilitating the fitting of models and subsequent evaluation using accuracy metrics.

At the outset of the experiment, the models were initially configured with rather high parameters, including a maximum word limit of 15,000 and a sequence length of 1,000. Regrettably, this aggressive configuration led to platform instability, ultimately causing system crashes after running the classifiers for a few hours.

Consequently, deemed it necessary to optimise the parameters for more sustainable computations. It was opted to reduce the maximum number of words to 800 and set a maximum sequence length of 100. This adjustment allowed for smoother and more reliable model training without challenging the system. Upon making this change and assessing the results after applying embedding techniques, it became apparent that with a maximum word limit set at 500, the coverage exceeded 100%. In the context of coverage, an ideal value falls between 0 and 1, where 1 signifies 100%. In pursuit of an optimal balance between computational efficiency and data coverage, further increased the maximum word limit to 800, which yielded a coverage of 89.867%. This level of coverage was deemed most suitable for the analysis, striking a balance between resource utilisation and data representation. The models MNB, MLP and SVM were performed with default parameters, and tuned with lower and higher parameters from the default, thus comparing the Precision, Recall, and F1-Score. In both the unprocessed and preprocessed experiments, all the developed models consistently demonstrated remarkable performance metrics, with scores ranging from 0.94 to 0.99 when utilising Vectorisation techniques, whereas the use of Embedding resulted in lower performance accuracy.

5. Results and Discussion

The results of the evaluation, conducted on two different datasets of "Fake and Real News," provide valuable insights into the performance of various models and feature selection methods.

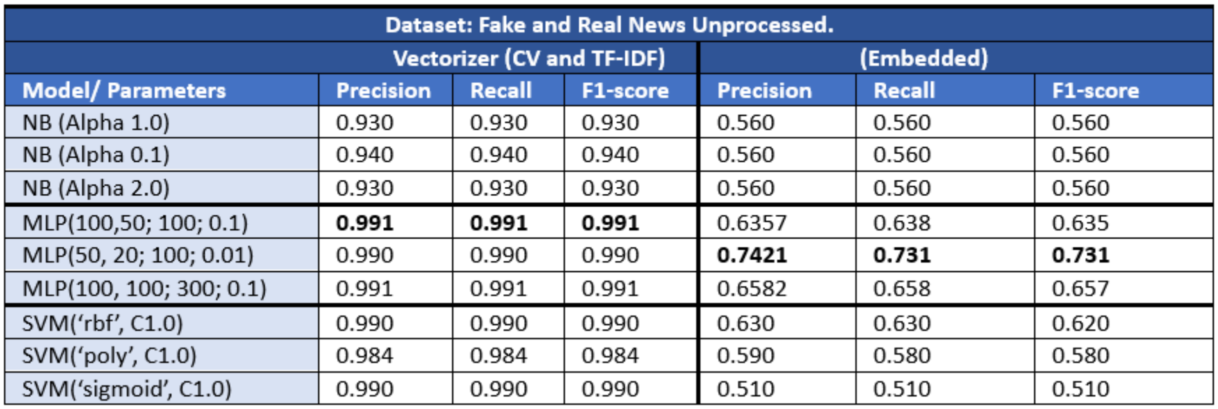
Unprocessed Dataset:

Following the implementation of three distinct model algorithms, an evaluation was performed using three evaluation metrics: Precision, Recall, and F1-score, and their corresponding parameters on the unprocessed dataset of “Fake and Real News", with two different selection methods: Vectorizer (Count Vectoriser and TF-IDF) and Embedded. The results, displayed in Figure 4 below, the most outstanding results are marked in bold.

1. Naive Bayes (NB): Regardless of the alpha value (1.0, 0.1, or 2.0) used for NB, the Precision, Recall, and F1-score remain consistently high at 0.93. However, these metrics drop significantly when using the Embedded feature selection method.

2. Multi-Layer Perceptron (MLP): MLP models, with various configurations, exhibit notable improvements in Precision, Recall, and F1-score compared to NB. The best-performing MLP model is MLP(100, 50; 100; 0.1) and (100, 100; 300; 0.1), achieving 0.991 for all three metrics when using the Vectorizer feature selection.

3. Support Vector Machine (SVM): SVM models with different kernels ('rbf', 'poly', 'sigmoid') and C values consistently demonstrate strong performance, with all metrics hovering around 0.990.



Dataset: Fake and Real News Unprocessed.						
	Vectorizer (CV and TF-IDF)			(Embedded)		
Model/ Parameters	Precision	Recall	F1-score	Precision	Recall	F1-score
NB (Alpha 1.0)	0.930	0.930	0.930	0.560	0.560	0.560
NB (Alpha 0.1)	0.940	0.940	0.940	0.560	0.560	0.560
NB (Alpha 2.0)	0.930	0.930	0.930	0.560	0.560	0.560
MLP(100,50; 100; 0.1)	**0.991**	**0.991**	**0.991**	0.6357	0.638	0.635
MLP(50, 20; 100; 0.01)	0.990	0.990	0.990	**0.7421**	**0.731**	**0.731**
MLP(100, 100; 300; 0.1)	0.991	0.991	0.991	0.6582	0.658	0.657
SVM('rbf', C1.0)	0.990	0.990	0.990	0.630	0.630	0.620
SVM('poly', C1.0)	0.984	0.984	0.984	0.590	0.580	0.580
SVM('sigmoid', C1.0)	0.990	0.990	0.990	0.510	0.510	0.510

Figure 2: Parameters and evaluation metrics for unprocessed data, using vectoriser and embedded features selection.

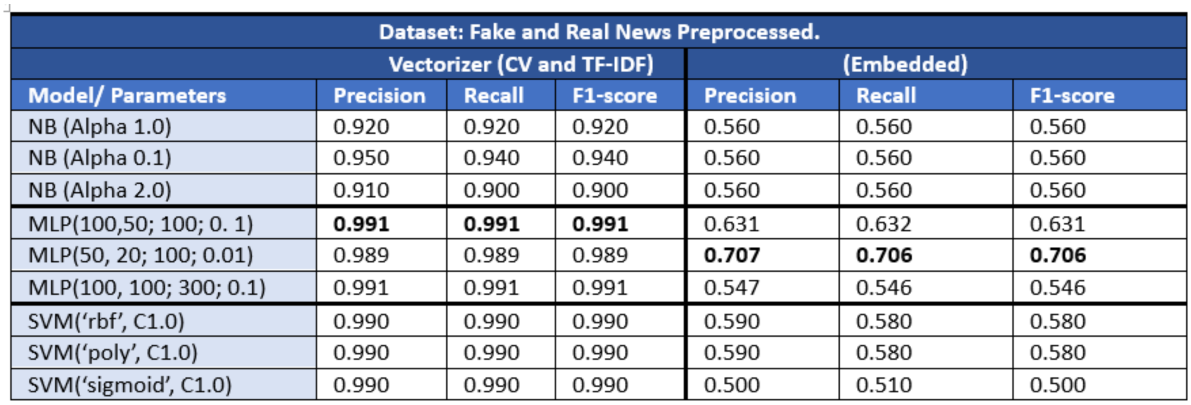
Preprocessed dataset:

Figure 5 illustrates the outcomes of model evaluation for the preprocessed 'Fake and Real News' dataset, following the same procedure as applied to the unprocessed dataset. The findings are as follow:

1. Naive Bayes (NB): Similar to the unprocessed dataset, NB exhibits consistent but lower performance across different alpha values, with all metrics at 0.920, and a more significant drop in performance when using the Embedded feature selection.

2. Multi-Layer Perceptron (MLP): MLP models continue to outperform NB, with the best results achieved with MLP is similar to unprocessed data, reaching 0.991 for Precision, Recall, and F1-score. However, the performance is generally lower than that in the unprocessed dataset, especially when using the Embedded feature selection.

3. Support Vector Machine (SVM): SVM models also show relatively strong performance but exhibit lower metrics compared to the unprocessed dataset, especially when employing the Embedded feature selection.



Dataset: Fake and Real News Preprocessed.						
	Vectorizer (CV and TF-IDF)			(Embedded)		
Model/ Parameters	Precision	Recall	F1-score	Precision	Recall	F1-score
NB (Alpha 1.0)	0.920	0.920	0.920	0.560	0.560	0.560
NB (Alpha 0.1)	0.950	0.940	0.940	0.560	0.560	0.560
NB (Alpha 2.0)	0.910	0.900	0.900	0.560	0.560	0.560
MLP(100,50; 100; 0. 1)	**0.991**	**0.991**	**0.991**	0.631	0.632	0.631
MLP(50, 20; 100; 0.01)	0.989	0.989	0.989	**0.707**	**0.706**	**0.706**
MLP(100, 100; 300; 0.1)	0.991	0.991	0.991	0.547	0.546	0.546
SVM('rbf', C1.0)	0.990	0.990	0.990	0.590	0.580	0.580
SVM('poly', C1.0)	0.990	0.990	0.990	0.590	0.580	0.580
SVM('sigmoid', C1.0)	0.990	0.990	0.990	0.500	0.510	0.500

Figure 3: Parameters and evaluation accuracy preprocessed data, using vectorizer and embedded features selection.

In summary, the findings indicate that MLP models consistently outperform Naive Bayes and SVM models, especially when employing the Vectorizer feature selection method, resulting in an impressive 99.1% precision, ensuring a high rate of correct predictions (true positives). This underscores MLP's capability to handle intricate, non-linear data. Furthermore, it's evident that the choice of dataset (unprocessed or preprocessed) and the feature selection method significantly impacts the model's performance. The Vectorizer method proves to be the more effective choice for this dataset, while the Embedded feature selection method yields lower metrics. Therefore, selecting the right feature selection approach and model configuration is of paramount importance in achieving the desired level of performance for the given task.

Conclusion and Reflections

In conclusion, the experiments and analysis have shed light on the influence of various factors, including algorithm types, parameters, pre-processing, and feature engineering techniques, on classification results. Based on the findings, it is propose the use of MLP and vectoriser (CountVectorizer and TF-IDF) for the 'Fake and Real News' dataset.

In theory, unprocessed data is expected to outperform preprocessed data due to the risk of overfitting in unprocessed data, where the model may capture noise, leading to poor generalisation. Preprocessing is designed to mitigate overfitting by cleaning and simplifying the data, allowing the model to focus on the most relevant information.
However, the empirical results challenge this theoretical expectation. Surprisingly, it was observed no significant differences in evaluation metrics between unprocessed and preprocessed data. This suggests that preprocessing may have limited impact in this context, or the data's quality may mitigate the effects of cleaning steps.

The research provides valuable insights into the complexities of data preparation and model selection in the context of these datasets, highlighting the need for further investigation and a nuanced approach to data processing in machine learning

Reflections

If I were to undertake the project again, I would implement several changes to enhance the overall process. Firstly, prioritise dedicating more time to a comprehensive review of the dataset and the documentation of potential limitations that might impact the experimental process. Dealing with low classification results led to the need for extensive backtracking to investigate the root cause of the problem, which was the data itself. Secondly, would place a stronger emphasis on the initial planning and scope definition. This would involve clearly defining project objectives and setting specific milestones, ensuring that the project's direction is well-structured right from the outset. Additionally, would adopt a more refined approach to feature engineering, exploring advanced techniques and transformations to extract more meaningful information from the data.

Moreover, the project provided a valuable learning experience, particularly in understanding how feature engineering can be tailored to meet the specific requirements of various model algorithms. This knowledge was instrumental in improving model performance and will be a valuable asset for future projects.

Reference

Akhtar, F., Khan, F. A., & Hanif, M. T. (2020). Fake and Real News Detection Using Python. International Journal of Scientific Research in Science and Technology, 7(3), 423. DOI: https://doi.org/10.32628/IJSRST207376

Amajd, M., Kaimuldenov, Z., & Voronkov, I. (2017, July). Text classification with deep neural networks. In International Conference on Actual Problems of System and Software Engineering (APSSE) (pp. 364-370).

Damaschk, M., Dönicke, T., & Lux, F. (2019, September). Multiclass text classification on unbalanced, sparse and noisy data. In Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing (pp. 58-65).

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. WSEAS transactions on computers, 4(8), 966-974.

Kadhim, A. I. (2018). An evaluation of preprocessing techniques for text classification. International Journal of Computer Science and Information Security (IJCSIS), 16(6), 22-32.

Macskassy, S. A., Hirsh, H., Banerjee, A., & Dayanik, A. A. (2003). Converting numerical classification into text classification. Artificial Intelligence, 143(1), 51-77.

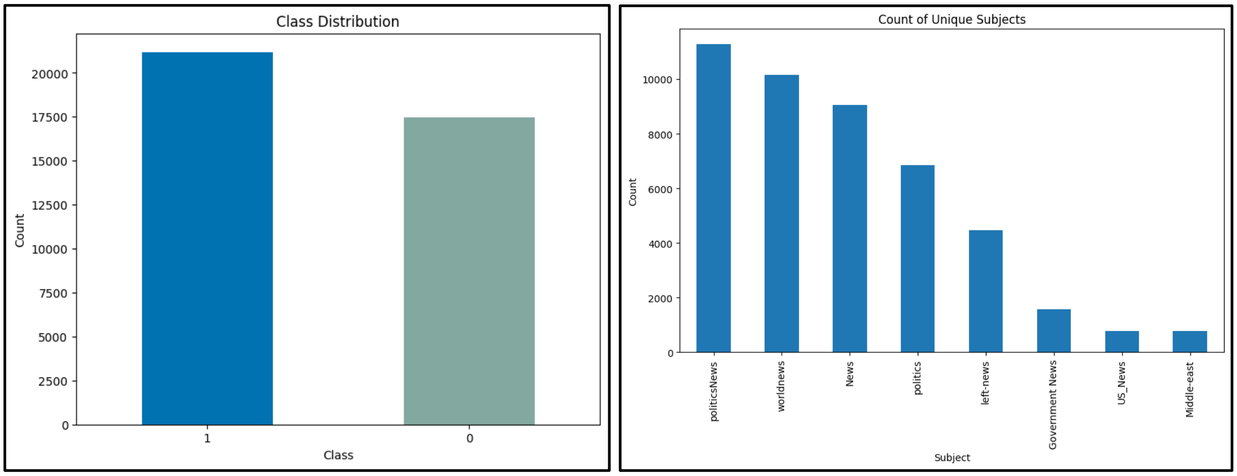
Singh, G., Kumar, B., Gaur, L., & Tyagi, A. (2019, April). Comparison between multinomial and Bernoulli naïve Bayes for text classification. In 2019 International Conference on Automation, Computational and Technology Management (ICACTM) (pp. 593-596).

Wendland, A., Zenere, M., & Niemann, J. (2021). Introduction to text classification: impact of stemming and comparing TF-IDF and count vectorization as feature extraction technique. In Systems, Software and Services Process Improvement: 28th European Conference, EuroSPI 2021, Krems, Austria, September 1–3, 2021, Proceedings 28 (pp. 289-300).

Appendices

Appendix A: Fake and Real News Dataset Graphs and Tables

A.1 - Target distribution, Fake (0) and True (1) news, and Count of Unique Subjects:


Class Distribution
Count
Class
Count of Unique Subjects
Count
politicsNews
worldnews
News
politics
left-news
Government News
US_News
Middle-east
Subject