News Classification using Machine Learning: A Comparative Study

Sabău Oana-Maria
Group 30232
Faculty of Automation and Computer Science
Sabau.Io.Oana@student.utcluj.ro

***Abstract*— This paper investigates the effectiveness of different machine learning algorithms for news classification. Specifically, it explores the performance of Multinomial Naive Bayes, Random Forest, and Linear Support Vector Classification (SVC) models on a news dataset. The study employs techniques such as text preprocessing, TF-IDF feature extraction, and grid search for hyperparameter optimization. Evaluation metrics including accuracy, precision, recall, and F1-score are utilized to assess the models' performance. The findings contribute to understanding the suitability of these algorithms for news categorization tasks.**

I. Introduction

In the rapidly evolving digital era, the volume of news articles produced daily is overwhelming. As information inundates various platforms, the need to efficiently categorize news articles has emerged as a fundamental requirement for effective information retrieval and analysis. News classification, which entails automatically assigning categories or labels to news articles based on their content, has thus become indispensable for organizing and comprehending the vast expanse of information accessible online.

A. Introduction to the Problem Statement: News Categorization Using Machine Learning

Despite the clear benefits of news classification, manually categorizing news articles is labor-intensive and impractical given the sheer volume of articles published daily. To address this challenge, machine learning techniques offer a promising solution by automating the process of categorizing news articles based on their textual content.

The problem statement of this study revolves around machine learning algorithms to develop accurate and efficient news classification systems. By training models on labeled datasets containing news articles and their corresponding categories and headlines, we aim to build predictive models capable of accurately categorizing unseen news articles into predefined categories.

B. Purpose of the Study and Research Objectives

The primary purpose of this study is to investigate the effectiveness of machine learning algorithms in automating the task of news categorization. Specifically, we seek to achieve the following research objectives:

- To explore the application of various machine learning algorithms, including Naive Bayes, Random Forest, and Linear SVC, in classifying news articles into predefined categories.
- To evaluate the performance of these machine learning models in terms of accuracy, precision, recall, and F1-score.
- To identify the optimal parameters for each machine learning algorithm through rigorous experimentation and hyperparameter tuning.
- To compare the effectiveness of different machine learning algorithms in news classification and determine which approach yields the best results.

C. Outline of the Paper

This paper is organized as follows:

1) **Introduction**: Provides an overview of news classification, introduces the problem statement, outlines the purpose of the study, and defines research objectives.
2) **Literature Review**: Reviews existing literature on news classification techniques and machine learning algorithms applied in text classification tasks.
3) **Data Preprocessing**: The dataset used in this study is described in detail. Data cleaning steps are explained, such as the replacement of certain categories and text preprocessing to remove irrelevant information.
4) **Feature Extraction**: The process of transforming textual data into numerical feature vectors using TF-IDF.
5) **Model Implementation**: Describes the implementation of three machine learning algorithms: Multinomial Naive Bayes, Random Forest, and Linear SVC. The models are trained on the training dataset and their performance is evaluated.
6) **Results**: Evaluation metrics such as accuracy, precision, recall, and F1-score are presented for each model. A comparison of the results obtained from the different classifiers is provided to determine the most effective model.
7) **Discussion**: The results are interpreted and a comparison between the classifiers is conducted.
8) **Conclusion**: The implications of the study for news classification applications are discussed, and suggestions for future research directions are provided.
9) **References**: Lists the references cited throughout the paper.

II. Literature Review

In this chapter, we review the existing body of research on news classification and discuss the machine learning algorithms relevant to text classification. We also summarize

the methodologies and findings of previous studies, providing a foundation for our research.

A. Existing Studies on News Classification

Text classification, also known as *Text Categorization*, is the method of categorizing and/or sorting the text-based entities into some predefined set of semantic categories or labels. There are various techniques that are being used in the process of automatic text classification, as presented in the *"Multi-category news classification using Support Vector Machine based classifiers"* research paper conducted by Saigal and Khanna [1], topic that will be addressed later in this section.

At the very first beginning, news classification has been a significant area of research in the field of Natural Language Processing (NLP). Various approaches have been employed to tackle the problem of categorizing news articles based on their content. Early works primarily focused on keyword-based methods and rule-based systems. *Natural Language Processing* studies interactions between system thinking and human languages, in the sense of how computer programs, process and analyses most natural language data. Using NLP, we classify textual data, fact described in detail by authors in the research work *Multi-Label News Category Text Classification* [2]. These NLP techniques were further replaced by more advanced and automatic method through the incorporation of Machine Learning Objectives in the task of automatic text classification, such as Naïve Bayes, a simple probabilistic model.

Another paper [3] suggests the working principle of a framework for text classification based on using KNN algorithm and TF-IDF method. *The K-Nearest Neighbor* (KNN) is one of the simplest lazy machine learning algorithms and its objective is to classify objects into one of the predefined classes of a sample group that was created by machine learning and *term frequency-inverse document frequency* (TF-IDF) is a numerical statistic method which allows the determination of weight for each term (or word) in each document. The method is often used in natural language processing (NLP) or in information retrieval and text mining and it features good results overall.

A very efficient method was approached by T. Joachims in his paper *"Text categorization with Support Vector Machines: Learning with many relevant features"*, [4] which implies that Support Vector Machine (SVM), a popular supervised learning algorithm used for pattern recognition and regression analysis, is a suitable choice for this categorization task.

Support Vector Machines (SVMs) are particularly well-suited for this task as they incorporate overfitting protection mechanisms that do not necessarily depend on the number of features, allowing them to handle these large feature spaces effectively. In text categorization, there are very few irrelevant features, making feature selection challenging. Experiments on the Reuters "acq" category indicate that even the lowest-ranked features, based on information gain, still contain valuable information. This suggests that aggressive feature selection may result in a loss of important information, and that an effective classifier should utilize many features to learn a "dense" concept. Joachims [1]-[4] concluded that the properties of text data—such as high dimensionality, sparse document vectors, and linear separability—combined with the characteristics of SVMs, provide strong theoretical evidence that SVMs should perform well in text categorization tasks.

The review of existing literature reveals that particular machine learning algorithms have been successful in the domain of text classification. The findings and methodologies of previous research provide a solid foundation for our study, which aims to further explore and compare these algorithms for news categorization.

III. Data Preprocessing

The data source is a Kaggle dataset [5] which contains around 210,000 news headlines from 2012 to 2022 from HuffPost [6]. Each record from the dataset consists of more attributes, but we only keep note of 'Category' and 'Headline'.

In the initial phase of our study, data preprocessing plays a crucial role in preparing the raw text data for subsequent analysis and model training. One of the primary tasks involved NLP techniques by standardizing the format of the headlines to ensure consistency across the dataset. The first step was to regroup categories with similar topics (e.g. Science and Tech, two separate labels, grouping into only one because it focuses on related aspects) going from 42 unique categories to 27 total. Next, we developed a custom text preprocessing function. This function encompassed several key tasks, including converting the text to lowercase. By converting all text to lowercase, we ensured that the machine learning models treat words with different cases (e.g., "Hello" and "hello") as the same, preventing redundancy and enhancing the efficiency of subsequent analyses.

Furthermore, the preprocessing function involved removing non-alphabetic characters from the headlines. This step aimed to eliminate noise and irrelevant symbols that could potentially interfere with the classification process. Additionally, tokenization was performed to break down the headlines into individual words or tokens. Tokenization facilitates the analysis of text data at a granular level, allowing the machine learning models to identify patterns and relationships more effectively.

Another crucial aspect of the preprocessing function was the removal of stopwords using the NLTK (Natural Language Toolkit) library's corpus. Stopwords are common words in a language (e.g., "the," "and," "is") that do not contribute significant meaning to the text. By filtering out stopwords, we focus the analysis on the most relevant and informative words in the headlines. Following the formulation of the preprocessing function, it was applied to the 'headline' column of our dataset.

The dataset is split into training and testing sets using the *train_test_split* function, the testing set size being set to 20% of the total dataset, and *random_state* is set to 42 for reproducibility.

IV. FEATURE EXTRACTION

In our study, feature extraction was a critical step that involved transforming the raw text data into numerical features that could be utilized by machine learning algorithms. The technique employed for this purpose was Term Frequency-Inverse Document Frequency (TF-IDF) vectorization.

Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus). [1]

Several parameters were fine-tuned to optimize the TF-IDF vectorization. The *max_features* parameter was set to 10,000 to control the maximum number of unique terms considered in the vectorization process. The *ngram_range* was specified as (1, 3), indicating that unigrams, bigrams, and trigrams were considered. This range allows the model to capture not only individual words but also contiguous sequences of two and three words, providing additional context and improving classification performance. The *max_df* parameter was set to 0.95, meaning that any term appearing in more than 95% of the documents would be ignored. Such terms are typically too common and do not carry discriminative information useful for classification. Conversely, the *min_df* parameter was set to 5, excluding terms appearing in fewer than 5 documents from the feature set.

In addition to transforming the text data, the target labels (categories) were encoded into numerical format using *LabelEncoder*, that was fitted to the training labels and subsequently used to transform both the training and test labels into integer values.

By converting both the textual data and category labels into numerical formats, a structured and standardized dataset was created, suitable for training and evaluating our machine learning models implemented. This feature extraction process was essential for leveraging the textual information present in the headlines and enabling effective news categorization.

V. MODEL IMPLEMENTATION

In this chapter, we discuss the implementation of the Machine Learning models chosen: Multinomial Naive Bayes, Random Forest, and Linear SVC.

Multinomial Naive Bayes (MNB) is a very popular and efficient machine learning algorithm that is based on Bayes' theorem. It is a probabilistic classifier to calculate the probability distribution of text data, which makes it well-suited for data with features that represent discrete frequencies or counts of events in various natural language processing (NLP) tasks.[8] To implement the Multinomial Naive Bayes classifier, we began by instantiating the *MultinomialNB* object from the sklearn.naive_bayes module. We defined a grid of parameters for the model's alpha hyperparameter, specifying values of 0.01, 0.1, 0.5, 1.0, and 2.0. To identify the optimal alpha value, we employed GridSearchCV, a hyperparameter tuning technique, conducting the grid search with 5-fold cross-validation and using accuracy as the scoring metric, issue that will be addressed in the next section.

To leverage the predictions of the Naive Bayes model as additional features, we first generated prediction probabilities for both the training and test datasets. These probabilities were then combined with the original TF-IDF features to create a new feature set for training a Random Forest classifier.

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks). [9] We instantiated the *RandomForestClassifier* with 100 estimators and trained it on the combined training set. The model's performance was evaluated on the combined test set.

Support Vector Classifier (SVC) is a specific implementation of the Support Vector Machine algorithm that is designed specifically for classification tasks. In other words, SVC is an SVM used for classification. It seeks to find the hyperplane that best separates the data points into different classes. The terms "SVC" and "SVM" are sometimes used interchangeably, but when someone refers to an "SVC," they are usually referring to the classification variant of the algorithm. [10] For the Linear Support Vector Classifier, we instantiated the *LinearSVC* object and a grid search was performed to tune the C parameter, exploring values of 0.01, 0.1, 1, 10, and 100. The optimal model identified by the grid search was then evaluated on the test data. Predictions were transformed back to their original labels for interpretation.

VI. RESULTS

The results for best parameter, accuracy, precision, F1-score and confusion matrix are included in the notebook and a further comparison is conducted.

The Multinomial Naive Bayes classifier was optimized using GridSearchCV to determine the best value for the alpha parameter. The grid search identified an optimal alpha value from the predefined range [0.01, 0.1, 0.5, 1.0, 2.0] and evaluated thay 0.1 is the most suitable value. Upon fitting the model with this optimal parameter, the classifier achieved an accuracy of **0.59** on the test set.

The Random Forest model, trained on the combined feature set, achieved an improved accuracy of **0.62** on the test set. This result underscored the benefit of leveraging predictions from the Naive Bayes model as supplementary features, but it took a longer time to complete.

The Linear Support Vector Classifier was also fine-tuned using GridSearchCV to find the optimal regularization parameter, C, from a specified range [0.01, 0.1, 1, 10, 100], implying that also 0.1 is the best parameter value. The optimized SVC model was evaluated on the test set, achieving an accuracy of **0.64**, slightly increased from the others.

VII. DISCUSSION

This section wraps up the topics discussed earlier and interprets and compares the results.

The models demonstrated notable performance in terms of accuracy. The *Multinomial Naive Bayes* classifier achieved an accuracy score of 0.59, notable for its rapid computation time of less than 50 seconds. The *Random Forest* classifier attained a slightly higher accuracy of 0.62, requiring a significantly longer training time of approximately 30 minutes. The *Linear SVC* model outperformed the others with an accuracy score of 0.64 and completed the task in about 10 minutes, making it both more accurate and relatively efficient.

For each model, the confusion matrix provided insightful details about their performance. Higher counts along the diagonal of the confusion matrices indicated accurate predictions, reflecting the models' strengths in correctly classifying certain categories. However, the matrices also revealed instances of misclassification where the models incorrectly assigned categories. This analysis highlighted specific areas where each model excelled or struggled, providing a deeper understanding of their strengths and weaknesses in the context of news categorization.

I attach each confusion matrix resulted to compare the results.

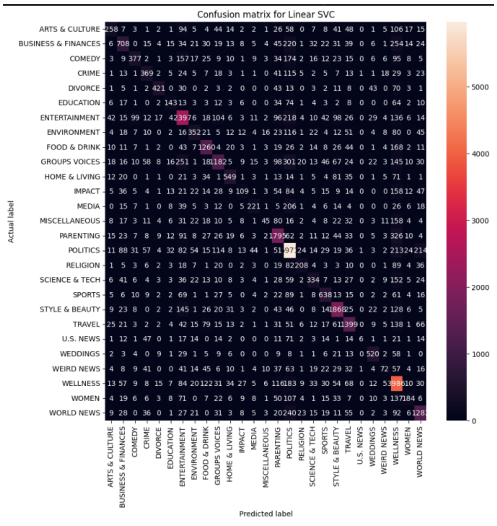

Fig. 1. Confusion Matrix for Linear SVC

A review of the existing literature underscores the effectiveness of Support Vector Machine (SVM) approaches in text classification tasks. Studies such as those referenced in [1] and [4] highlight SVM's ability to generalize well in high-dimensional feature spaces, reducing the need for extensive feature selection and thus simplifying the text categorization process. Consistent with these findings, our *Linear SVC* model demonstrated superior accuracy and reasonable computation time, even with the substantial volume of data involved. This aligns with the established advantages of SVMs in handling complex, high-dimensional datasets efficiently.

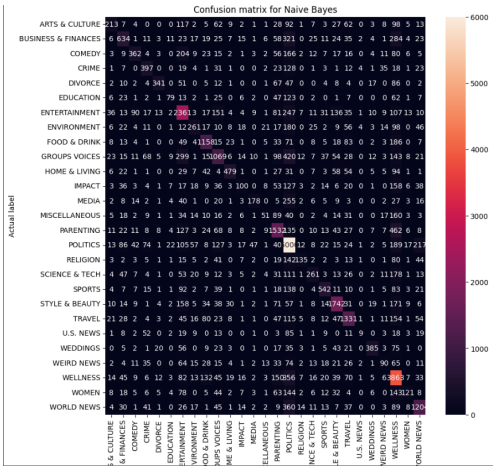

Fig. 2. Confusion Matrix for Naive Bayes

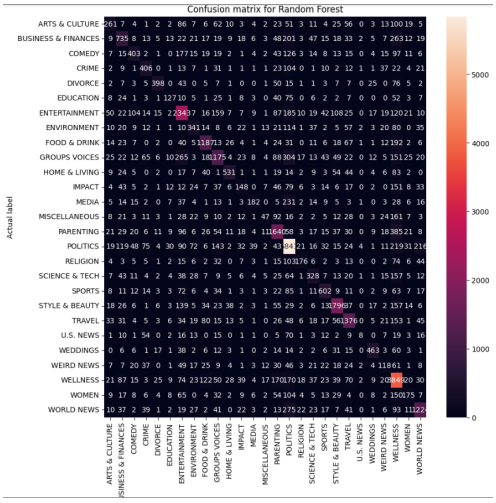

Fig. 3. Confusion Matrix for Random Forest

VIII. CONCLUSION

This study highlights the potential and limitations of three popular machine learning models for news classification. Each model presents distinct advantages depending on the specific requirements of the application, such as the need for speed versus accuracy. The Multinomial Naive Bayes classifier, with its rapid computation time, is preferable for real-time news categorization where quick response times are essential. On the other hand, applications that can afford longer processing times but require higher accuracy might benefit more from using Linear Support Vector Classifier (SVC) or Random Forest classifiers.

The analysis revealed that while all models performed well on certain categories, there were noticeable misclassifications, indicating areas for improvement.

Future research could explore hybrid models that combine the strengths of different classifiers, such as integrating the rapid predictions of Naive Bayes with the higher accuracy of SVC. Additionally, leveraging more powerful machine learning and deep learning tools, such as XGBoost and neural

networks, could further advance the field. Developing and testing these models in real-time news categorization systems to evaluate their performance in live environments would also be valuable.

By addressing the identified challenges and exploring the suggested research directions, future work can significantly advance the field, leading to more accurate and efficient news classification systems. This progress will be crucial for developing robust and reliable news categorization tools that can effectively handle the diverse and dynamic nature of news content.

REFERENCES

[1] Saigal, P., Khanna, V. "Multi-category news classification using Support Vector Machine based classifiers." SN Appl. Sci. 2, 458 (2020). Retrieved from https://doi.org/10.1007/s42452-020-2266-6

[2] Shilpa Patil, V. Lokesha, Anuradha S. G., "Multi-Label News Category Text." JOURNAL OF ALGEBRAIC STATISTICSVolume 13, No. 3, 2022, p.5485-5498. Retrieved from https://publishoa.com/index.php/journal/article/view/1417

[3] Bruno, T., Sasa, M., Donko, D., "KNN with TF-IDF based framework for text categorization", 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013. Retrieved from https://www.researchgate.net/publication/269688447_KNN_with_TF-IDF_based_framework_for_text_categorization

[4] Joachims, "T. Text categorization with Support Vector Machines: Learning with many relevant features." Nédellec, C., Rouveirol, C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer, Berlin, Heidelberg. Retrieved from https://link.springer.com/chapter/10.1007/BFb0026683

[5] Kaggle News Category Dataset. Retrieved from https://www.kaggle.com/datasets/rmisra/news-category-dataset/code

[6] HuffPost. Retrieved from https://www.huffpost.com/

[7] Karabiber, F. "TF-IDF — Term Frequency-Inverse Document Frequency". Learn Data Sci. https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/

[8] Multinomial Naive Bayes. Retrieved from https://www.geeksforgeeks.org/multinomial-naive-bayes/

[9] Random Forest Algorithm in Machine Learning. Retrieved from https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

[10] Salunke, D. SVC "(Support Vector Classifier)", 2023. Retrieved from https://www.linkedin.com/pulse/svc-support-vector-classifier-dishant-salunke/