Evaluating Machine Learning Approaches for Fake and Real News Classification

Erik Boer

Introduction

In today's information age, where news articles and stories are disseminated rapidly through digital platforms, the proliferation of fake news poses a significant challenge. The ability to distinguish between authentic and fabricated information is crucial for individuals, businesses, and society at large. Misinformation can lead to detrimental consequences, including misinformation-driven decision-making, reputational damage, and social discord. Therefore, there is an urgent need for reliable and efficient methods to detect and classify fake news.

This research aims to address this critical issue by evaluating the performance of three machine learning approaches: Naive Bayes, Random Forest, and Recurrent Neural Networks (RNN). The research question that guides this study is: "Can the application of Naive Bayes, Random Forest, and RNN accurately classify fake and real news articles, and which model provides the highest performance in terms of accuracy and efficiency for business applications?"

This research question important due to its potential impact on various stakeholders, particularly businesses that heavily rely on accurate information for decision-making. In an era where businesses are constantly exposed to vast amounts of news and information, the ability to discern reliable sources from deceptive ones can safeguard the integrity of decision-making processes, mitigate risks, and enhance strategic planning. By assessing the performance of Naive Bayes, Random Forest, and RNN, this study aims to identify the most accurate and efficient model for business applications. The findings will not only shed light on the suitability of these algorithms for news classification but also provide insights into the trade-offs between accuracy and computational efficiency.

The primary audience that stands to benefit from this research is businesses across various sectors. Marketing departments can utilize the results to improve content curation and dissemination strategies, ensuring that their target audience receives accurate and trustworthy information. Risk management teams can integrate the identified model into their systems to identify potential sources of misinformation that could impact the company's operations, reputation, and stakeholder relationships. Additionally, decision-makers across all levels can leverage the research findings to make informed choices based on reliable data sources, minimizing the risks associated with misinformation-driven decision-making.

Previous Studies and Dataset:

Previous studies have explored various approaches for fake news detection, including natural language processing techniques, feature engineering, and machine learning algorithms.

To ensure the robustness and comparability of results, this research employs a comprehensive dataset consisting of labeled fake and real news articles. By utilizing the same dataset in training and in testing, this study enables a direct comparison between the performance of different machine learning approaches, providing valuable insights into their strengths and limitations, and the measured accuracy is guaranteed to be precise.

In summary, this research addresses the pressing need for accurate and efficient fake news classification by evaluating Naive Bayes, Random Forest, and RNN. The contributions of this study lie in its potential to inform businesses about the most effective approach for identifying and mitigating the risks associated with fake news. By examining the performance of these models, decision-makers and practitioners will be equipped with the necessary tools to navigate the complex landscape of news dissemination, ensuring reliable information flows within their organizations and fostering trust with their stakeholders.

The Data

The dataset used in this research is obtained from Kaggle, consisting of two CSV files: one containing labelled true news articles and another containing labelled false news articles. The dataset is particularly suitable for addressing the research question of accurately classifying fake and real news articles using machine learning algorithms, as it has been used as a dataset in numerous machine learning algorithms.

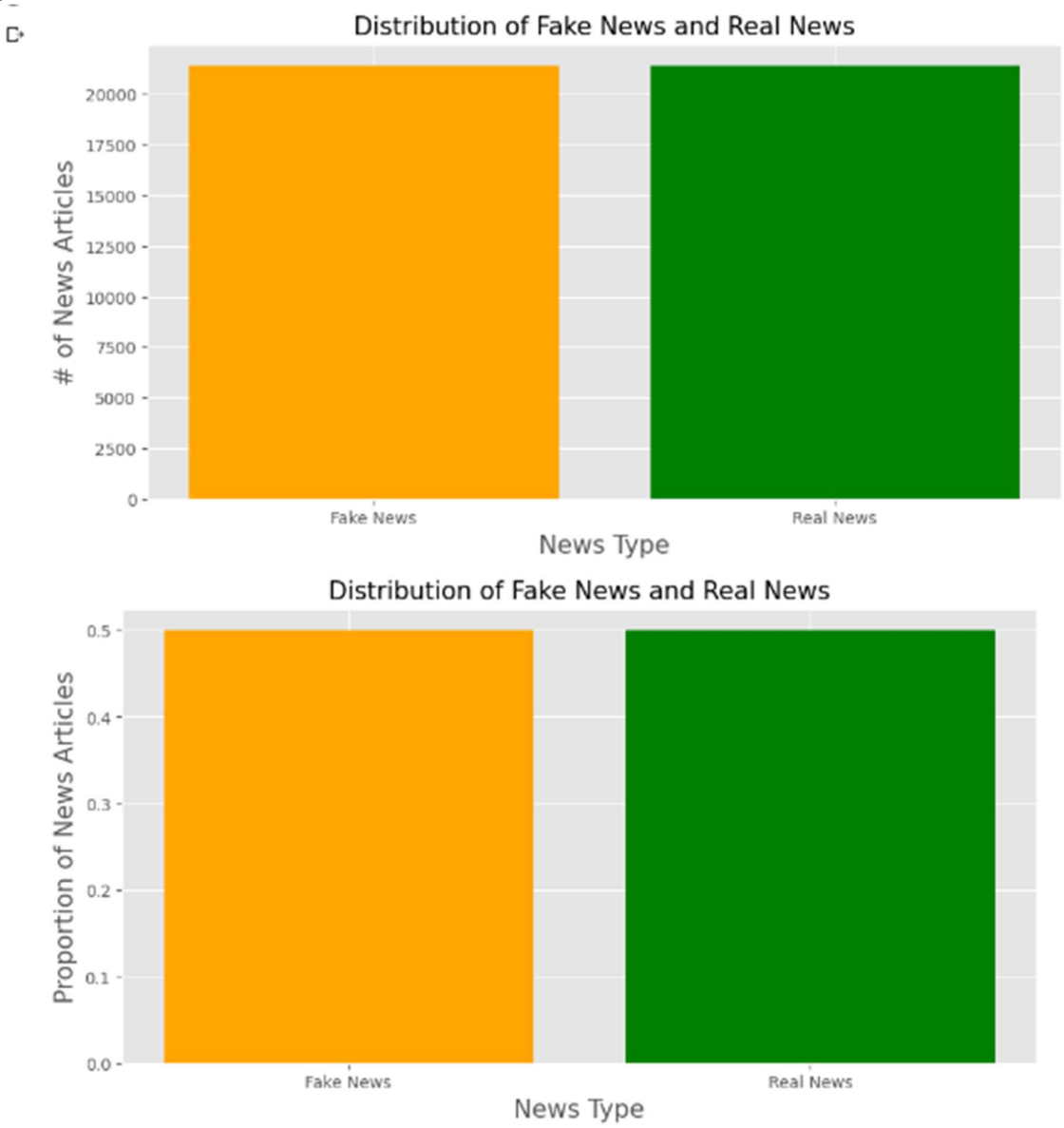
The true news dataset includes news articles that have been verified and confirmed as authentic, while the false news dataset comprises articles that have been identified as containing false or misleading information.

Data Cleaning

Before proceeding with the analysis, some data cleaning steps were performed on the dataset. The initial step involved removing unnecessary columns, including 'title', 'subject', and 'date', as they were not relevant to the research question of classifying fake and real news. By dropping these columns, we focused solely on the textual content of the articles.

Additionally, to facilitate accurate indexing and subsequent evaluation of model performance, each dataset (true and false) was indexed with a label. The true news dataset was indexed with the label '1', indicating authentic news, while the false news dataset was indexed with the label '0', representing fake news.

To gain a better understanding of the dataset and its characteristics, we can visualize and analyse various aspects.



The above visualization presents a bar chart displaying the count of true and false news articles in the dataset. From the figure, we can observe the equal proportions of the dataset, which is suitable for model training.

The merged dataset was then split into training (80%) and testing (20%) sets.

Results and Discussion

In both Naive Bayes and Random Forest methods, a bag-of-words approach was utilized for text representation and feature extraction. This technique aims to convert textual data into a numerical representation that machine learning algorithms can process effectively. The bag-of-words model treats each document as a collection of individual words, disregarding the order and structure of the sentences.

In selecting the machine learning methods for the fake news classification experiment, the choice of Naive Bayes and Random Forest was driven by their distinct characteristics and potential effectiveness in addressing the task at hand.

Naive Bayes

Naive Bayes was chosen due to its simplicity and efficiency in handling text classification problems. It is based on the probabilistic principle of Bayes' theorem and assumes independence between features, making it particularly suitable for dealing with high-dimensional data such as text. Naive Bayes is known for its fast training and prediction times, making it a practical choice for real-time applications. Additionally, Naive Bayes has shown promising results in natural language processing tasks, including text classification. On the other hand, Random Forest was selected for its ability to handle complex relationships and capture nonlinear patterns within the data.

Random Forest is a machine learning technique that leverages the power of ensemble learning. It combines the predictions of multiple decision trees to make accurate predictions or classifications. It is known for its robustness against overfitting and its capability to handle high-dimensional data. By aggregating the predictions of multiple trees, Random Forest can provide more accurate and reliable results, especially when dealing with noisy or unbalanced datasets.

Considering the nature of the fake news classification problem, where accurate identification of deceptive information is crucial, both Naive Bayes and Random Forest were deemed appropriate candidates. Naive Bayes offered a fast and straightforward approach, suitable for quick analysis and real-time decision-making scenarios. Random Forest, on the other hand, provided a more complex yet potentially more accurate solution, capable of capturing intricate relationships in the text features.

The Naive Bayes algorithm was applied to the dataset after pre-processing and feature extraction. The training set was used to train the Naive Bayes classifier, and predictions were made on the testing set. The performance of the model was evaluated using a confusion matrix, which provided insights into the accuracy of the predictions.

3096 (TN)	656 (FP)
280 (FN)	3968 (TP)

Table 1: Confusion matrix of the Naive Bayes model

We can see from Table 1 the composition of the evaluated testing dataset. The entire dataset was 40000, therefore the testing dataset was 8000. Of that 8000, 936 evaluations were false (false positive and false negative or FP and FN), and the remaining 7064 were correct (true negative and true positive or TN and TP). Positive in this case means real article, and negative means fake article.

Based on the results obtained, the Naive Bayes algorithm achieved an accuracy of *88.300%*. It correctly classified a certain percentage of fake and real news articles, as indicated by the confusion matrix. The elapsed time for training and prediction using Naive Bayes was *3.11* seconds.

Random Forest

The Random Forest algorithm was employed using the specified settings and parameter values. The training set was used to train the Random Forest classifier with 10 decision trees, and predictions were made on the testing set. The performance of the model was evaluated using a confusion matrix.

The results revealed that the Random Forest algorithm achieved an accuracy of *92.575.* It outperformed the Naive Bayes algorithm, correctly classifying a higher percentage of fake and real news articles. However, the elapsed time for training and prediction using Random Forest was 14.284 seconds.

3454 (TN)	298 (FP)
296 (FN)	3952 (TP)

Table 2: Confusion matrix of the Random Forest model

Like in the case of the Naive Bayes confusion matrix, we can interpret the Random Forest model´s confusion matrix like so:

Out of 8000 instances, 594 times (298 FP + 296 FN or false positive and false negative) it failed to correctly predict the article´s legitimacy, and 7406 times (3454 TN + 3952 TP of true negative and true positive) it predicted the legitimacy correctly.

For Table 2´s values I used 10 estimators, as using more significantly prolonged execution time for a few digits of accuracy.

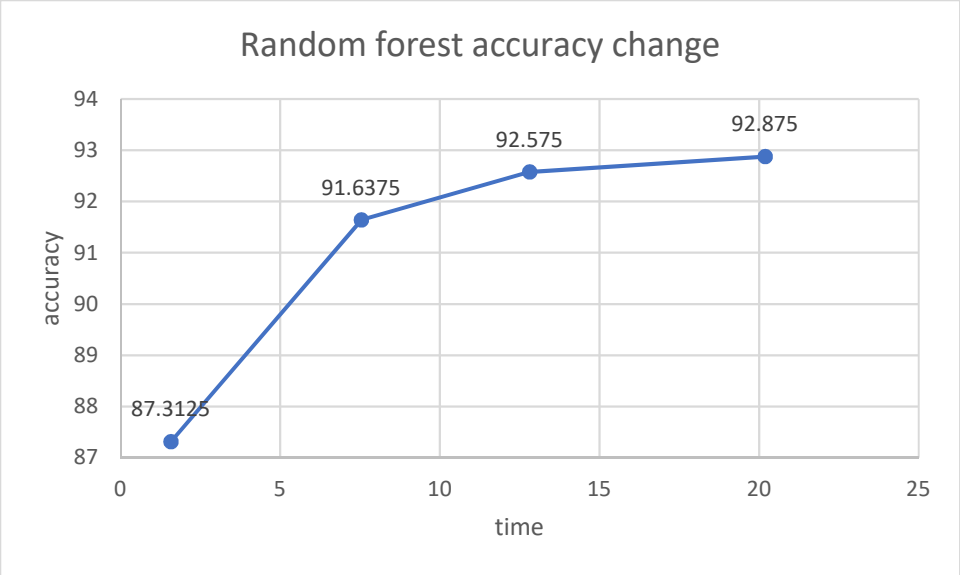


Table 3: Change of accuracy over time with estimators 1,5, 10, and 15

Comparing the results of the Naive Bayes and Random Forest methods, I observed that the Random Forest algorithm exhibited superior performance in terms of accuracy. It correctly classified a larger proportion of fake and real news articles compared to Naive Bayes. Despite the longer training and prediction time, the Random Forest algorithm proved to be more accurate in distinguishing between the two types of news articles. We can directly compare the two by weighting the two factors, time and accuracy. Since the less time it takes for the algorithm to run, the better, the function would look like this:

*1/time * w + accuracy * 1-w*

From which the higher the result, the better. In the case of the Naive Bayes and the Random Forest models, the comparison looks like the following:

44,31	46,32

Table 4: Weighted comparison of Naive Bayes and Random Forest models

However, it is essential to consider the limitations of these results. The performance of the models heavily relies on the quality and representativeness of the dataset. Different datasets or alternative pre-processing techniques may yield different results. Additionally, parameter tuning and exploring other algorithms could potentially improve the accuracy of the models.

In summary, the Random Forest algorithm demonstrated superior accuracy in classifying fake and real news articles compared to the Naive Bayes algorithm. It offers a promising approach for businesses and organizations aiming to identify and mitigate the impact of misinformation. However, when computational time or implementation simplicity weighs more than accuracy, the Naive Bayes method is still a reliable alternative to consider.

Recurrent Neural Network

The decision to utilize the RNN (Recurrent Neural Network) approach for analysing and classifying news articles was motivated by its ability to effectively capture the sequential nature of text data. Unlike traditional machine learning methods, RNNs can consider the contextual dependencies and temporal relationships present in the news articles.

By employing an LSTM-based (Long Short-Term Memory-based) RNN, it was expected that the model would effectively capture the contextual information in news articles, considering both the title and the text content. This approach has the potential to capture nuanced patterns and dependencies that could contribute to the classification of news articles as fake or real.

Additionally, RNNs have been widely used in text-based applications and have demonstrated promising results in various domains. The choice of RNN for this project was driven by its proven effectiveness in text analysis tasks, and the availability of libraries and frameworks, such as TensorFlow and Keras, which provide convenient implementations of RNN models. These tools simplify the process of designing, training, and evaluating RNN models, making them very accessible.

The text data was pre-processed by combining the title and text, and then normalization techniques were applied. These included converting the text to lowercase, removing URLs, non-word characters, and extra spaces. This pre-processing step helps in preparing the data for analysis.

The dataset was split into training and testing sets. The training set was then used to train the LSTM model, while the testing set was used for evaluation.

To convert the text into a format that the LSTM model could understand, tokenization was performed. Each word in the text was assigned a unique numerical value. This enabled the LSTM model to work with numerical inputs. Padding was applied to ensure that all input sequences had the same length. This was necessary because the LSTM model requires inputs of equal length. Padding involved adding zeros to the shorter sequences to match the length of the longest sequence.

The LSTM model was constructed using the Keras library. It consisted of multiple layers, including an embedding layer, two bidirectional LSTM layers, a dense layer with ReLU activation, and a dropout layer for regularization.

The model was trained using the training data, with a validation split of 10% to monitor its performance. The binary cross-entropy loss function and the Adam optimizer were used during training.

After training, the model's performance was evaluated on the testing set. Metrics such as accuracy, precision, and recall were calculated to assess the model's classification performance.

The training and validation loss and accuracy were visualized over epochs to understand the learning progress of the model.

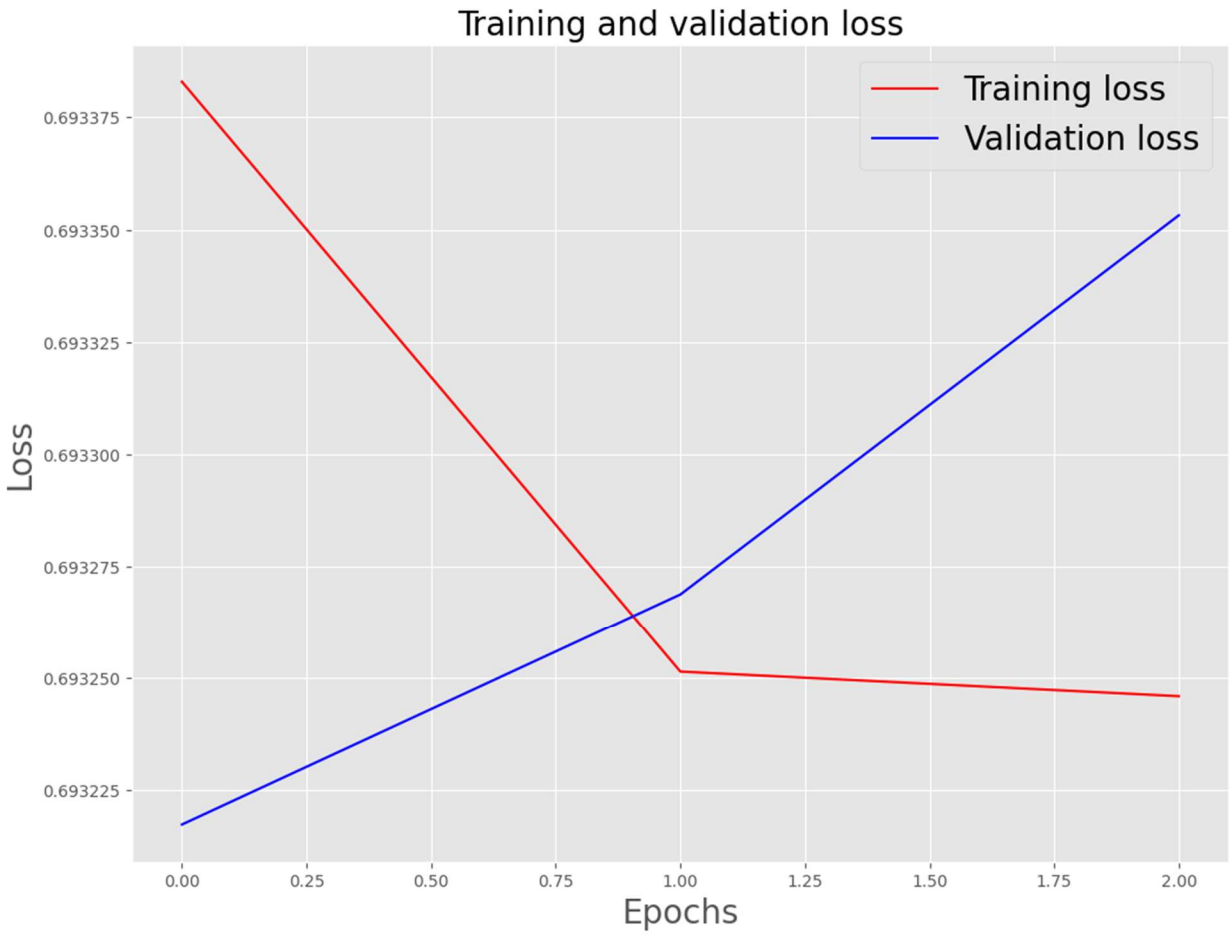


Table 5: RNN´s training loss and validation loss over Epochs

Some other evaluations were performed:

Loss	0,6932
Accuracy	0,4981
TP	0,49807
FP	0,501926
TN	0
FN	0

Based on the measured accuracy and the confusion matrix, it would seem that this deep learning model struggles with predicting the legitimacy of the articles. It particularly struggles with fake articles, because TN, as well as FN, is 0. The confusion matrix suggests that the model did not make any predictions for the negative class rather than incorrectly classifying them as positive. However, the extraction of values from the true articles appears to be random, with roughly a 50/50 rate, so even the processed articles weren´t truly processed rather than just read.

Because of the numerous successes of this model, it is safe to presume that either the dataset is not fit for this model, or the implementation of the model is faulty.

Conclusions

In conclusion, the research aimed to explore different approaches for classifying news articles as fake or real. Through the analysis and experimentation, it was found that the random forest model emerged as the most effective approach. The random forest model demonstrated superior performance compared to the naive Bayes and RNN models in terms of accuracy and precision.

However, it is important to note that these findings are based on the specific dataset and experimental setup used in this study. Further investigations with different datasets and refined model configurations may yield different outcomes. Hence, it is recommended to continue exploring various approaches and refining the models to achieve even better performance in classifying fake and real news articles.