Fake News Detection Using a Logistic Regression Model and Natural Language Processing Techniques

Johnson Adeleke Adeyiga
Bells University of Technology
Philip Gbounmi Toriola
Bells University of Technology
Temitope Elizabeth Abioye(Ogunbiyi) (elizatope_2005@yahoo.com)
Bells University of Technology
Adebisi Esther Oluwatosin
Bells University of Technology
oluwasefunmi 'Tale Arogundade
Federal University of Agriculture

Research Article

Keywords: Fake News, Term Frequency &ndash; Inverse Document Frequency (TF-IDF), Logistic Regression, Natural Language Processing, Python

Posted Date: July 14th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3156168/v1

Abstract

The proliferation of fake news has become a significant challenge in recent years, impacting democracy, the journalism industry, and people's daily lives. The spread of intentionally misleading or fabricated information has led to a decline in confidence in government institutions and has profound implications for people's daily lives. This study aims to detect false information and real news using logistic regression algorithms and natural language processing techniques, implement the model using Python, and develop a website for news classification. The "Fake News Detection" dataset from Kaggle, consisting of approximately 20,000 news articles labelled as real or fake, was used. Data cleaning was done and feature extraction techniques, including Term Frequency – Inverse Document Frequency (TF-IDF) vectorization, was applied to extract features from the data. The logistic regression model with K-Nearest Neighbour (KNN), Passive Aggressive classifier and Naïve Bayes model were trained on the extracted features and evaluated using various metrics. The system was implemented using Python and Google Collaboratory, with the front-end of the website developed using Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. The architecture of the system involves training the model and deploying it using Flask, a lightweight web framework. Evaluation of the classifiers show the following results: Accuracy - 97.90%, 82.92%, 91.32%, 89.71%; Precision - 96.59%, 80.25%, 94.13%, 93.88%; Recall - 99.32%, 94.08%, 90.88%, 89.14%; and F1 score - 97.94%, 86.62, 92.48, 91.45% for logistic regression, KNN, Passive Aggressive and Naïve Bayes classifier respectively. Based on these results, logistic regression outperformed the other three classifiers. This shows that logistic regression model is more effective in fake news detection. The developed system provides a valuable tool in combating fake news and contributes to the on-going research in automatic fake news detection using machine learning.

1. INTRODUCTION

Fake news has emerged as a significant problem for democracy, the journalism industry, and the economy [1]. This deliberate dissemination of verifiably erroneous information poses a threat to public confidence in government institutions and significantly impacts people's daily lives [2]. The rise of fake news is not a recent phenomenon, as even before the internet era, newspapers would often manipulate information to serve their own agendas [1]. This practice eroded trust in the media and created a culture of scepticism. However, with the advent of the internet and social media, the spread of fake news has reached unprecedented levels. The ease of sharing information on social media platforms has made it easier for individuals to propagate false or misleading information to a wide audience [3]. Moreover, the emergence of fake news websites and social media accounts has made it increasingly challenging for individuals to distinguish between legitimate and fraudulent sources.

The consequences of fake news are far-reaching. It can manipulate public opinion, distort democratic processes, and foster the spread of misinformation. The 2016 US Presidential election serves as a stark example, where fabricated news stories were used to influence the outcome, disseminating false claims about candidates [4]. This misinformation undermines trust in genuine news sources and threatens the foundations of democracy.

Furthermore, fake news can have detrimental effects on public health. Misinformation about the safety and efficacy of vaccinations, for instance, can lead to decreased vaccination rates and increased spread of infectious diseases. Similarly, false information about disease causes and treatments can misguide individuals, resulting in the adoption of ineffective or harmful remedies.

Addressing the challenge of fake news detection has become paramount in curbing the proliferation of false information across various mediums, including social media and messaging platforms [5]. Fact-checking has been a widely used method, involving the manual verification of news articles against credible sources [6]. Media literacy and critical thinking skills have also been promoted to empower individuals to critically evaluate news stories and identify biases and fallacies [7], [8]. However, these approaches have limitations, particularly in coping with the vast volume of information generated every minute.

To overcome these limitations, researchers are turning to machine learning, a powerful technology that enables computers to learn from data and make decisions autonomously [9]. Machine learning algorithms offer promise in detecting fake news, as they can be trained on labelled datasets consisting of known fake and legitimate news articles. This enables the algorithm to classify new articles as fake or legitimate automatically. While various classifiers have been utilized, this research focuses on employing Logistic Regression fused with Natural Language Processing techniques to enhance fake news detection.

In this work, a fake news dataset was obtained from Kaggle, consisting of approximately 20,000 news articles labelled as real or fake and was merged into a CSV file. This dataset was cleaned using pre-processing steps and features were extracted using TF-IDF vectorizer. The vectorized features were fed into the algorithm for training and evaluation. The text model is split into 80% and 20% for training and testing respectively. The Logistic Regression machine learning model was used and a fake news classification model was built taking the vector generated from the TF-IDF vectorizer as input for training and evaluation. The results obtained from the model are as follows: Accuracy – 98.65% on training data and 97.90% on test data, Precision – 96.59%, Recall – 99.32% and F1 score – 97.94%. This was done using Python on the Google Collab platform. The results obtained with logistic regression model was benchmarked using three other machine learning classifiers – K-Nearest Neighbour (KNN), Passive Aggressive classifier and Naïve Bayes classifier. The evaluation results for these three models were: Accuracy – 82.92%, 91.32%, 89.71%; Precision – 80.25%, 94.13%, 93.88%; Recall – 94.08%, 90.88%, 89.14%; and F1 score – 86.62, 92.48, 91.45% respectively. This results show that logistic regression is more effective in fake news detection.

By implementing this model, a fake news detection website was developed where users can input news content and it will be classified as real or fake. The front-end is developed using HTML, CSS and JavaScript and the website's prediction code is integrated with the front-end through the use of a Python module called Flask, which constantly runs in the background to enable computations on the site.

2. RELATED WORKS

In the information age, addressing the issue of fake news has been a topic of high relevance. Researchers have conducted various studies with different perspectives, approaches, and techniques to counteract misinformation. Here are some notable contributions:

The study by Agudelo *et al.*,[10] presented a novel approach for detecting false news using machine learning algorithms, natural language processing, and Python programming. By utilizing CountVectorizer, TfidfVectorizer algorithms, and a Multinomial Naive Bayes model, they were able to achieve high classification accuracy of 88.1% and 84.8%, respectively, on a dataset consisting of over 10,000 news items. However, ethical implications of automated fake news detection were not discussed.

Yang *et al.*, [11] proposed the TI-CNN model for identifying fake news in online social networks, which outperformed several baseline methods with impressive accuracy of 92.20%. However, their evaluation was limited to a dataset collected before the 2016 US presidential election.

In their study, Patel *et al.*, [12] discussed the use of Natural Language Processing techniques along with various classifiers to detect fake news. They emphasized the importance of early identification to prevent the spread of fake news. SGD algorithm proved to be the most effective with an accuracy of 90.32% for constant learning rate and 90.42% for adaptive learning rate. SVM and KNN also gave good results with an accuracy of 88.47% and 86.90% respectively, while K-means had a very low accuracy of fake news prediction with 40.37%.

The study conducted by Srinivasa Rao *et al.*, [13] introduced a fake news detection approach using the BERT model. They highlighted the significance of context sensitivity in creating accurate models and demonstrated superior performance compared to logistic regression and K-Nearest Neighbours.

Vikram Tembhurne & Almin [14] proposed a Multichannel Deep Neural Network (McDNN) model for fake news detection, surpassing existing techniques. They showcased the potential applications of the McDNN model in various industries. The study showed that the McDNN outperformed the state-of-the-art machine learning and deep learning techniques for fake news detection with an accuracy of 94.68% on Fake News Data (FND) and an accuracy of 99.23% on the ISOT Fake News Dataset.

Kulkarni *et al.*, [5] addressed the concern of fake news in online resources by proposing a machine learning-based detection model. Their study employed classifiers like Random Forest, Logistic Regression, Decision Tree, KNN, and Gradient Booster, the results showed that Logistic Regression achieved the highest accuracy of 85.04%, followed by Random Forest with 84.50% accuracy. The KNN algorithm and Decision Tree achieved 80.20% and 78.11% accuracy, respectively, while Gradient Boosting algorithm had the lowest accuracy of 77.44%.

In another research by Ajao *et al.*, [15] developed an approach combining CNN and LSTM models to detect and classify fake news on Twitter. While achieving promising accuracy, the approach may lack contextual understanding for complex fake news stories.

An investigation on fake news detection using machine learning techniques, with Logistic Regression performing the best was performed by Kushwaha & Singh [16]. They also developed a web-based deployment of the model. However, the study acknowledged the need for a more comprehensive dataset.

3. METHODOLOGY

The development of the model involved several key steps. A dataset containing real and fake news articles was obtained from Kaggle and supplemented with additional articles from various sources. The data was cleaned and pre-processed to make it suitable for analysis, including tasks like removing punctuation, tokenization, eliminating stop words, stemming, and POS tagging. Feature extraction was performed using the TF-IDF vectorizer to convert the text into numerical representations. A logistic regression model with KNN, Passive Aggressive classifier and Naïve Bayes classifier were trained on the extracted features using Python libraries. The models' performance were evaluated using metrics such as True Positive, True Negative, False Positive, and False Negative. The system was implemented using Python programming language in Google Colab, with a front-end developed using HTML, CSS, and JavaScript.

The study way run and tested on a DELL Latitude 5480 with Intel(R) Core(TM) i5-7300U CPU at 2.70 GHz, 7.88 GB RAM size available and 54 GB disk size available. Flask, a lightweight web framework, was used to develop the web application for deploying the trained model.

3.1 Dataset

Fake news dataset on Kaggle called the "Fake News Detection" dataset will be used, which contains approximately 20,000 news articles labelled as either "real" or "fake". The dataset also includes articles that were collected from various sources including Politifact, Buzzfeed, and Snopes. The dataset includes metadata such as the article's headline, author, and text body. The data set obtained was merged into a CSV file for processing.

To understand the dataset better, tokens of the news content was generated, producing a corpus of the text. These tokens were then plotted on a word cloud based on the frequency of occurrence of each word. This was done by utilizing the 'wordcloud' and 'matplotlib' tools on Python.

3.2 Text Cleaning

For each of the news content in the dataset, the following text pre-processing steps were taken:

1. Lowercasing the text: This step converted all the text to lowercase letters. It is done to ensure consistency and to make it easier to cross-check words with stopword and pos_tag dictionaries. By converting everything to lowercase, the potential discrepancies caused by words appearing in different cases is eliminated.

2. Removing words with only one letter: This step removed words that consist of only one letter. In many cases, such words do not carry much meaning and are not informative for analysis. Removing them helps to reduce noise and focus on more relevant words.
3. Removing words that contain numbers: This step removed words that contain any numerical characters. Similar to the previous step, words with numbers are often not meaningful in the context of textual analysis and can be safely discarded.
4. Tokenizing the text and removing punctuation: Tokenization is the process of splitting text into individual words or tokens. By tokenizing the text, it is broken down into its constituent parts for further analysis. Additionally, punctuation marks are removed during this step as they are not informative for most natural language processing tasks.
5. Removing all stop words: Stop words are commonly used words that do not carry much meaning in a given context, such as articles (e.g., "the", "a"), prepositions (e.g., "in", "on"), and conjunctions (e.g., "and", "but"). Removing these stop words helped to reduce noise in the text and allowing focus on more content-bearing words.
6. Removing empty tokens: After tokenization, there may be instances where empty tokens are generated. These empty tokens do not contribute to the analysis and are removed to ensure that only meaningful words are considered.
7. POS tagging the text: POS (Part-of-Speech) tagging is the process of classifying words into their respective parts of speech, such as nouns, verbs, adjectives, adverbs, etc. This step provided additional information about the role and grammatical function of each word, which can be useful for further analysis and understanding the text. The NLTK library provides a pos_tag() function that can be used to perform POS tagging.
8. Lemmatizing the text: Stemming and lemmatization were the techniques used to reduce words to their base or root form. Stemming involved removing prefixes and suffixes, while lemmatization mapped words to their base or dictionary form. These techniques help to normalize the words and reduce redundancy, allowing for better analysis and comparison of the text. For example, lemmatizing the words "running" and "runs" would result in both being reduced to the base form "run". This helps to reduce the dimensionality of the data and improve the accuracy of subsequent analyses.

3.3 Feature Extraction

Feature extraction is the process of simplifying and transforming raw data into a more manageable set of important characteristics, making it easier for machine learning models to understand and analyse the data. By creating representations of words that capture their meanings, semantic relationships, and diverse contextual uses, computers are able to understand text and perform tasks like clustering and classification. In this work, TF-IDF vectorizer was used to extract features from the text.

3.3.1 Term Frequency – Inverse Document Frequency (TF-IDF) Vectorizer

When words are scored based on their frequency alone, problem such as, common words tending to have higher scores even though they may not provide much meaningful information to the model were encountered. On the other hand, rare words that are specific to a particular subject or domain may contain more valuable information. To address this issue, the approach used was to adjust the word frequency by considering how often they appear across all documents. By doing so, scores of frequent words that are common across all documents were penalized. This scoring approach is known as Term Frequency – Inverse Document Frequency (TF-IDF).

The TF component which stands for Term Frequency as seen in Eq. 1, of TF-IDF calculates how often a term appears in a document, taking into account the varying document sizes.

$$TF\left(t,d\right)=\frac{Number of times t occurs in document \prime d\prime}{Total word count of document \prime d\prime}(1)$$

In Eq. 2, the IDF component of TF-IDF is the Inverse Document Frequency, which downplays the significance of commonly used words such as "a", "an", "the", "on", and "of". IDF increases the importance of uncommon words, and the greater the value of IDF, the more unique the word.

$$IDF\left(t,d\right)=\frac{Total number of documents}{Number of documents with the term t in it}(2)$$

TF-IDF in Eq. 3 is typically employed to analyse the body text and capture the relative frequency of each word within the sentences, which is then stored in the document matrix.

$$TF-IDF\left(t,d\right)=TF\left(t*d\right)*IDF\left(t\right)(3)$$

The vectorized features were fed into the algorithm for training and evaluation. The text model is split into 80% and 20% for training and testing respectively.

3.4 Building and Training the Machine Learning Model

The Logistic Regression model, benchmarked with K-Nearest Neighbour, Passive Aggressive classifier and Naïve Bayes classifier, was used in this work. From this machine learning algorithm, a fake news classification model was built taking the vector generated from the TF-IDF vectorizer as input for training and evaluation.

3.4.1 Mechanisms and Processes of Operation of the Model

Logistic regression is a statistical model used to analyse the relationship between input features and a binary outcome variable. In this case, the input features represent characteristics of a news article, and the outcome variable indicates whether the article is real or fake. Let's denote the input features as X, where X = [$x_1, x_2, \ldots, x_n$], and the outcome variable as y, where y takes on binary values of either 0 (indicating a real news article) or 1 (indicating a fake news article). The logistic regression model aims to estimate the probability that an article is fake (y = 1) given the input features X. The logistic regression model estimates the probability of an article being fake based on the input features. It uses a logistic or sigmoid function to map the linear combination of the input features to a value between 0 and 1. The logistic function is defined according to Eq. 4:

$$sigmoid\left(z\right)=\frac{1}{1+e^{-z}}(4)$$

Here, z represents the linear combination of the input features and their associated coefficients. Mathematically, z is calculated according to Eq. 5:

$$z=b_0+b_1x_1+b_2x_2+b_3x_3+\cdots+b_nx_n\left(5\right)$$

In this equation, $b_0, b_1, b_2, \ldots, b_n$ represents the coefficients or weights associated with each input feature $x_1, x_2, \ldots, x_n$. The coefficients are learned during the training phase of the logistic regression model. During the training phase, the model figures out the coefficients for each input feature by reducing a loss function. After determining these coefficients, the model can predict the outcome variable for new articles by evaluating the probability of it being 1 based on the input features. The sigmoid function is used to calculate this probability. It takes the value of z, obtained by combining the input features with their respective coefficients, and transforms it into the predicted probability. This probability as seen in Eq. 6 represents how certain the model is about classifying the article as fake.

$$P\left(y=1|\,X\right)=sigmoid\left(b_0+b_1x_1+b_2x_2+b_3x_3+\cdots+b_nx_n\right)\left(6\right)$$

By comparing the predicted probability to a threshold, which 0.5 for this work, the model classifies the article as either real or fake.

3.5 Model Evaluation

These evaluation metrics are used to measure how well the classification model performs. They give specific measurements that help to understand the accuracy and effectiveness of the model. The metrics used include True Positive (TP), which represents instances correctly predicted as positive, True Negative (TN), which represents instances correctly predicted as negative, False Positive (FP), which represents instances incorrectly predicted as positive, and False Negative (FN), which represents instances incorrectly predicted as negative.

Using these values, the following can be evaluated:

1. Accuracy: Accuracy calculated the overall correctness of the classifier by dividing the number of correctly classified instances (True Positives and True Negatives) by the total number of instances in the dataset. It is represented by the Eq. 7:

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}(7)$$

2. Precision: It measured the proportion of correctly predicted positive instances (True Positives) out of the total instances predicted as positive (True Positives and False Positives). It evaluates the classifier's ability to minimize false positives and is calculated as seen presented in Eq. 8:

$$Precision = \frac{TP}{TP + FP} (8)$$

3. Recall (Sensitivity or True Positive Rate): Recall determined the proportion of correctly predicted positive instances (True Positives) out of the total actual positive instances (True Positives and False Negatives). It measures the classifier's ability to identify all positive instances and is given by Eq. 9:

$$Recall = \frac{TP}{TP + FN} (9)$$

4. F1 Score: The F1 score is a metric that combined precision and recall into a single measure. It is the harmonic mean of precision and recall, providing a balanced assessment. The F1 score is especially useful when there is an uneven distribution of classes in the dataset. It can be calculated using Eq. 10:

$$2 \bullet \frac{Precision \bullet Recall}{Precision + Recall} (10)$$

3.6 Website Development

The implementation of the system utilized the Python programming language and Google Colab. Google Colab, a cloud-based platform offered by Google, enables collaborative writing, running, and sharing of Python code. It provided an interactive environment specifically designed for working with Jupyter notebooks. Jupyter Notebook is a widely utilized web-based IDE that is popular among data scientists and researchers for Python development. The front-end is developed using HTML, CSS and JavaScript and the website's prediction code is integrated with the front-end through the use of a Python module called Flask, which constantly runs in the background to enable computations on the site.

3.6.1 Website’s Working Mechanism

Flask, a lightweight web framework, was used to develop the web application for deploying the trained model. The Flask application handles incoming requests from users and provides responses based on the classification results. The Flask app loads the trained model and vectorizer used during training. When a user submits a text for classification, the input text is pre-processed and transformed using the trained vectorizer. The pre-processed text is then passed through the trained model, which predicts whether the text is real or fake news. The classification result is returned as a response to the user through the website interface.

4. RESULTS AND DISCUSSION

4.1 RESULTS

After evaluating the classifiers, the result obtained is presented in Table 1. The following results were obtained for logistic regression: Accuracy – 98.65% on training data and 97.90% on test data, Precision – 96.59%, Recall – 99.32% and F1 score – 97.94%. With this result, logistic regression is more effective in fake news detection and chosen as the preferred model for the system development.

Table 1
Evaluation Results of the Classifiers

Metric	**Logistic Regression %**	**K-Nearest Neighbour %**	**Passive Aggressive %**	**Naïve Bayes %**
Accuracy	97.90	82.92	91.32	89.71
Precision	96.59	80.25	94.13	93.88
Recall	99.32	94.08	90.88	89.14
F1 Score	97.94	86.62	92.48	91.45

Table 2
Confusion matrix for Logistic Regression Classifier

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive – 2004	False Positive – 73
Actual Negative	False Negative – 14	True Negative – 2069

This means that of the 4160 samples that were tested, the classifier correctly identified 2004 samples as "fake news"; correctly identified 73 samples as "real news"; incorrectly classified 2069 samples as "fake news" when they were actually "real news" and incorrectly classified 14 samples as "real news" when they were actually "fake news."

To visualize the performance and gain further insights, the following visualizations were created:

4.1 Discussion

From Fig. 5, the ROC curve shows that the classifier has good performance as it is closer to the top-left corner. A curve closer to the diagonal (random classifier) indicates poorer performance.

In Fig. 6, the precision-recall curve, positioned closer to the top right corner, indicates a classifier that is highly effective in distinguishing between real and fake news. This demonstrates the classifier's accuracy in making precise predictions and capturing a substantial portion of the positive samples.

The learning curve, depicted in Fig. 7, offers valuable information about the model's performance. A high training accuracy suggests that the model successfully captures the patterns present in the training data, indicating its ability to learn from the provided examples. Additionally, high validation accuracy suggests that the model performs well on unseen or validation data. The relatively small gap between the training and validation accuracy indicates that the model generalizes well and avoids significant overfitting. However, if there is a noticeable difference between the two accuracies, it suggests that there is room for improvement in the model's generalization performance.

To contextualize the results obtained, [5] reported an accuracy of 85%, precision of 84%, recall of 87%, and F1 score of 85% for their logistic regression classifier. Also, [12] reported an accuracy of 90.32% with SDG and 88.47% and 86.90% with SVM and KNN respectively. The result obtained from the classifier developed in this work surpasses these results which indicate the effectiveness of the logistic regression model and natural language techniques used.

5. CONCLUSION

Based on the analysis done, it was observed that logistic regression model achieved high accuracy, precision, recall, and F1 score in classifying news content as real or fake when compared with K-Nearest Neighbour (KNN), Passive Aggressive classifier and Naïve Bayes classifier. This suggests that the selected features, the natural language techniques used and the logistic regression algorithm were effective in distinguishing between real and fake news. Hence, within the limit of the data and results obtained from this work, it can be concluded that:

- This text classification system effectively distinguished between real and fake news articles, showcasing high accuracy, precision, recall, and F1 score.
- This project demonstrates the potential of machine learning techniques in combating the spread of misinformation and assisting in information verification.
- With further enhancements and the inclusion of other languages, the model could be extended to address broader challenges in the realm of news classification and verification.

Declarations

DECLARATION OF INTEREST

There is no conflict of interest among the authors. All authors contributed equally to the successful implementation of the work.

FUNDING

This research work has no funding.

References

1. P. Goyal, S. Taterh, and A. Saxena, “Fake News Detection using Machine Learning: A Review,” *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, vol. 7, no. 3, pp. 2454–1311, 2021, doi: 10.22161/ijaems.
2. X. Zhou, R. Zafarani, K. Shu, and H. Liu, “Fake News: Fundamental theories, detection strategies and challenges,” *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pp. 836–837, Jan. 2019, doi: 10.1145/3289600.3291382.
3. S. Hakak, W. Z. Khan, S. Bhattacharya, G. T. Reddy, and K. K. R. Choo, “Propagation of Fake News on Social Media: Challenges and Opportunities,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12575 LNCS, pp. 345–353, 2020, doi: 10.1007/978-3-030-66046-8_28.
4. H. Allcott and M. Gentzkow, “Social Media and Fake News in the 2016 Election,” *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, Mar. 2017, doi: 10.1257/JEP.31.2.211.
5. P. Kulkarni, S. Karwande, R. Keskar, P. Kale, and S. Iyer, “Fake News Detection using Machine Learning,” *ITM Web of Conferences*, vol. 40, p. 03003, 2021, doi: 10.1051/itmconf/20214003003.
6. H. Allcott, M. Gentzkow, and C. Yu, “Trends in the diffusion of misinformation on social media,” *Research and Politics*, vol. 6, no. 2, Apr. 2019, doi: 10.1177/2053168019848554.
7. S. M. Jones-Jang, T. Mortensen, and J. Liu, “Does Media Literacy Help Identification of Fake News? Information Literacy Helps, but Other Literacies Don’t,” *American Behavioral Scientist*, vol. 65, no. 2, pp. 371–388, Feb. 2021, doi: 10.1177/0002764219869406.
8. P. Machete and M. Turpin, “The Use of Critical Thinking to Identify Fake News: A Systematic Literature Review,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12067 LNCS, pp. 235–246, 2020, doi: 10.1007/978-3-030-45002-1_20.
9. A. Raj, “A Review on Machine Learning Algorithms,” *Int J Res Appl Sci Eng Technol*, vol. 7, no. 6, pp. 792–796, Jun. 2019, doi: 10.22214/IJRASET.2019.6138.
10. G. Agudelo, O. Parra, and J. Barón Velandia, “Raising a Model for Fake News Detection Using Machine Learning in Python,” pp. 596–604, 2018, doi: 10.1007/978-3-030-02131-3_52ï.
11. Y. Yang *et al.*, “TI-CNN: Convolutional Neural Networks for Fake News Detection,” Jun. 2018, Accessed: Mar. 24, 2023. [Online]. Available: https://arxiv.org/abs/1806.00749v3
12. J. Patel, M. Barreto, U. Sahakari, and Dr. S. Patil, “Fake News Detection with Machine Learning,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 10, no. 1, pp. 124–127, Nov. 2020, doi: 10.35940/IJITEE.A8090.1110120.
13. D. Srinivasa Rao, N. Rajasekhar, D. Sowmya, D. Archana, T. Hareesha, and S. Sravya, “Fake News Detection Using Machine Learning Technique,” in *SCRS CONFERENCE PROCEEDINGS ON INTELLIGENT SYSTEMS*, Soft Computing Research Society, 2021, pp. 59–69. doi: 10.52458/978-93-91842-08-6-5.
14. J. Vikram Tembhurne and M. M. Almin, “Mc-DNN: Fake News Detection Using Multi-Channel Deep Neural Networks,” *Int J Semant Web Inf Syst*, vol. 18, no. 1, doi: 10.4018/IJSWIS.295553.

15. O. Ajao, D. Bhowmik, and S. Zargari, “Fake news identification on Twitter with hybrid CNN and RNN models,” 2018, doi: 10.1145/3217804.3217917.
16. N. S. Kushwaha and P. Singh, “Fake News Detection using Machine Learning: A Comprehensive Analysis,” *Journal of Management and Service Science (JMSS)*, vol. 2, no. 1, pp. 1–15, Feb. 2022, doi: 10.54060/JMSS/002.01.001.

Figures

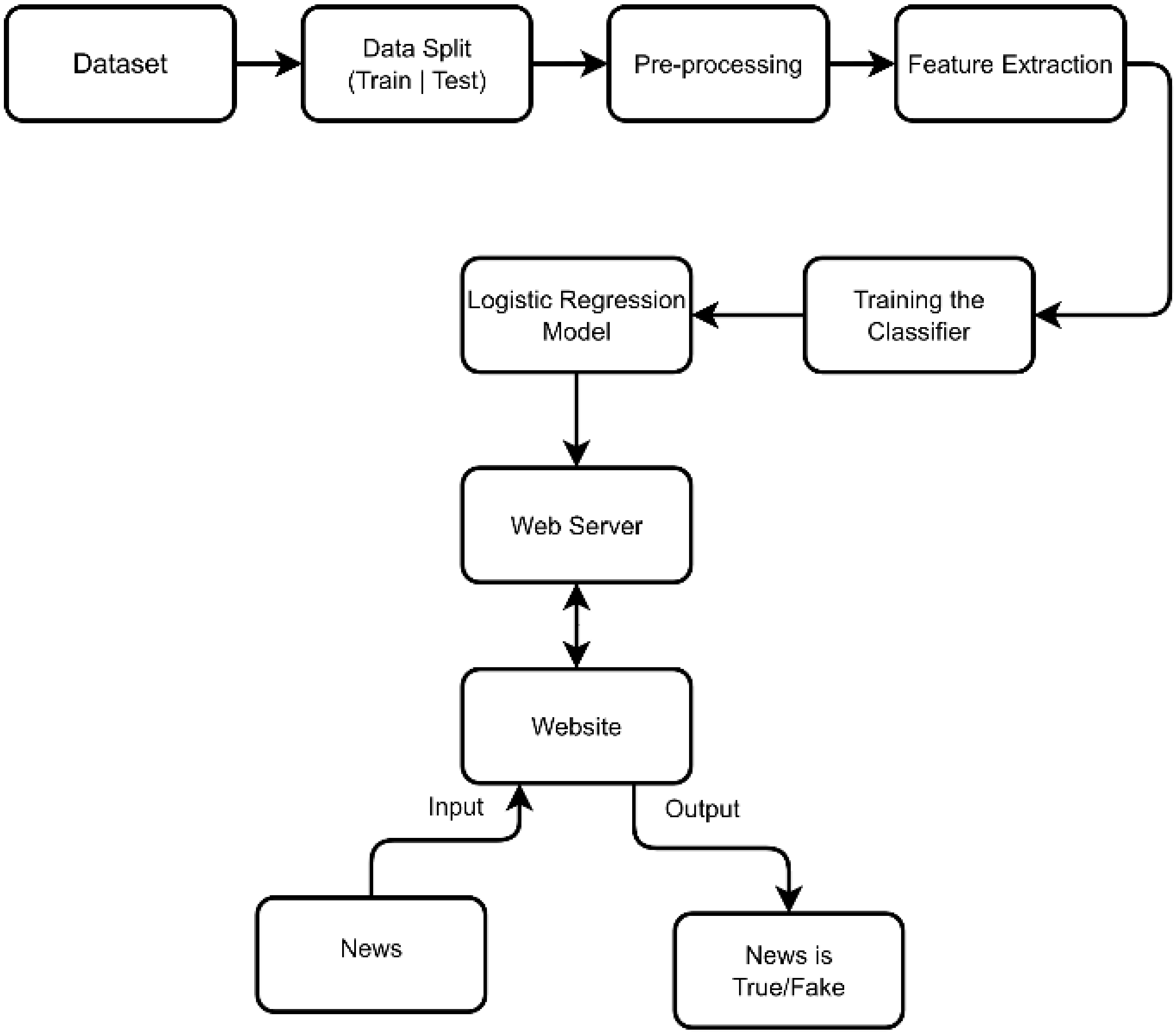


Figure 1

Workflow Diagram

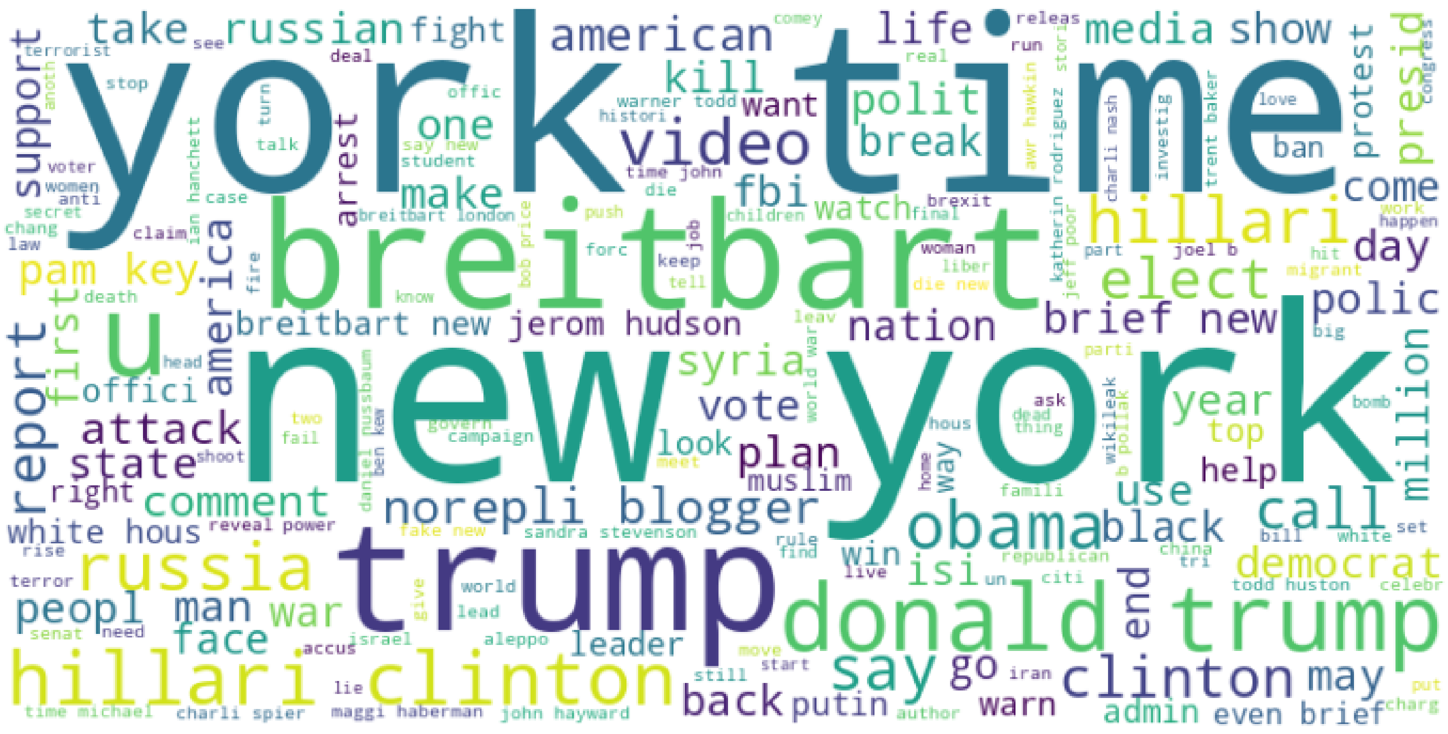
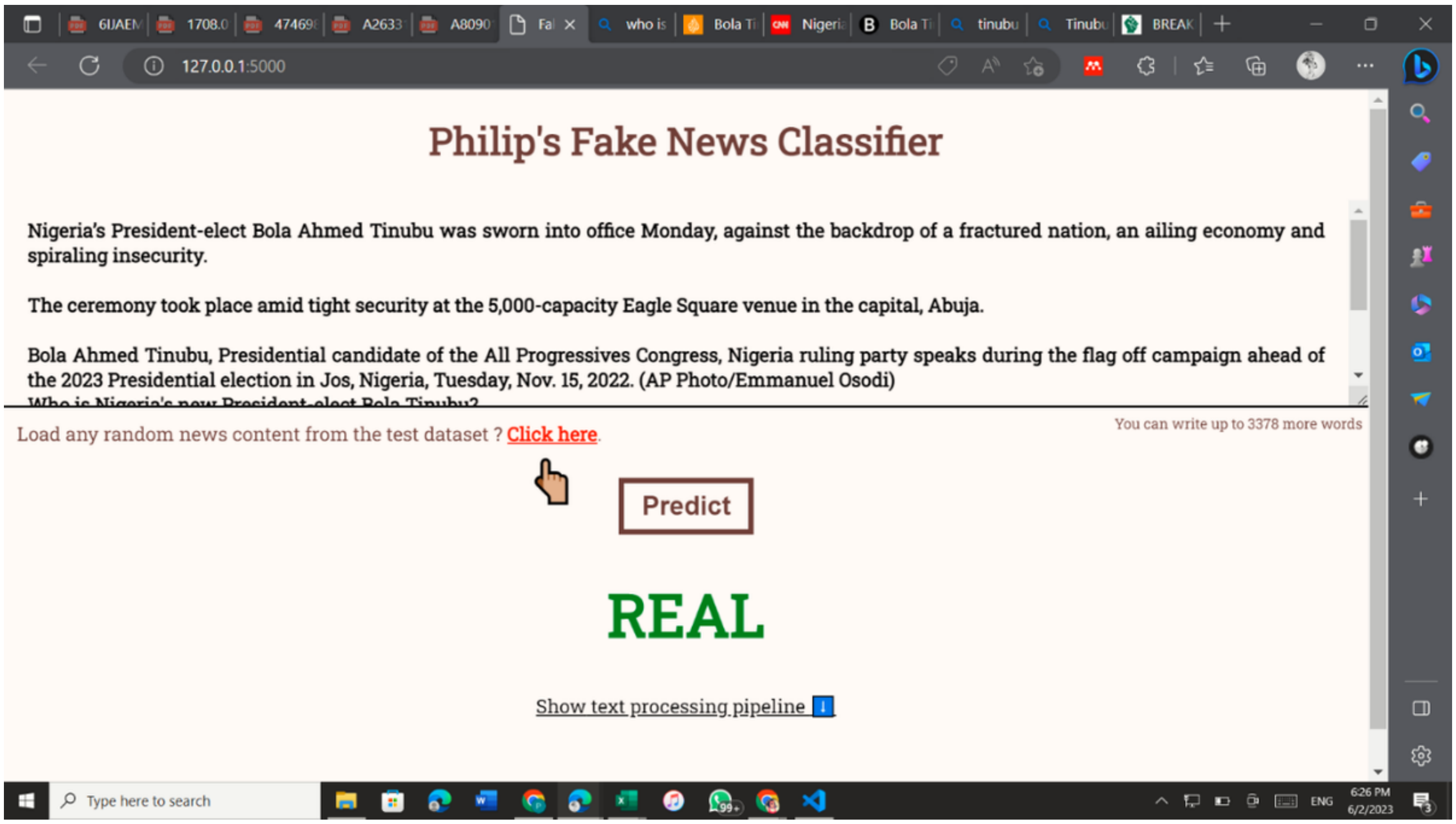


Figure 2

Word Cloud Plot



Philip's Fake News Classifier

Nigeria's President-elect Bola Ahmed Tinubu was sworn into office Monday, against the backdrop of a fractured nation, an ailing economy and spiraling insecurity.

The ceremony took place amid tight security at the 5,000-capacity Eagle Square venue in the capital, Abuja.

Bola Ahmed Tinubu, Presidential candidate of the All Progressives Congress, Nigeria ruling party speaks during the flag off campaign ahead of the 2023 Presidential election in Jos, Nigeria, Tuesday, Nov. 15, 2022. (AP Photo/Emmanuel Osodi)

Who is Nigeria's new President-elect Bola Tinubu?

Load any random news content from the test dataset ? Click here.

You can write up to 3378 more words

Predict

REAL

Show text processing pipeline

Figure 3

User interface of the website

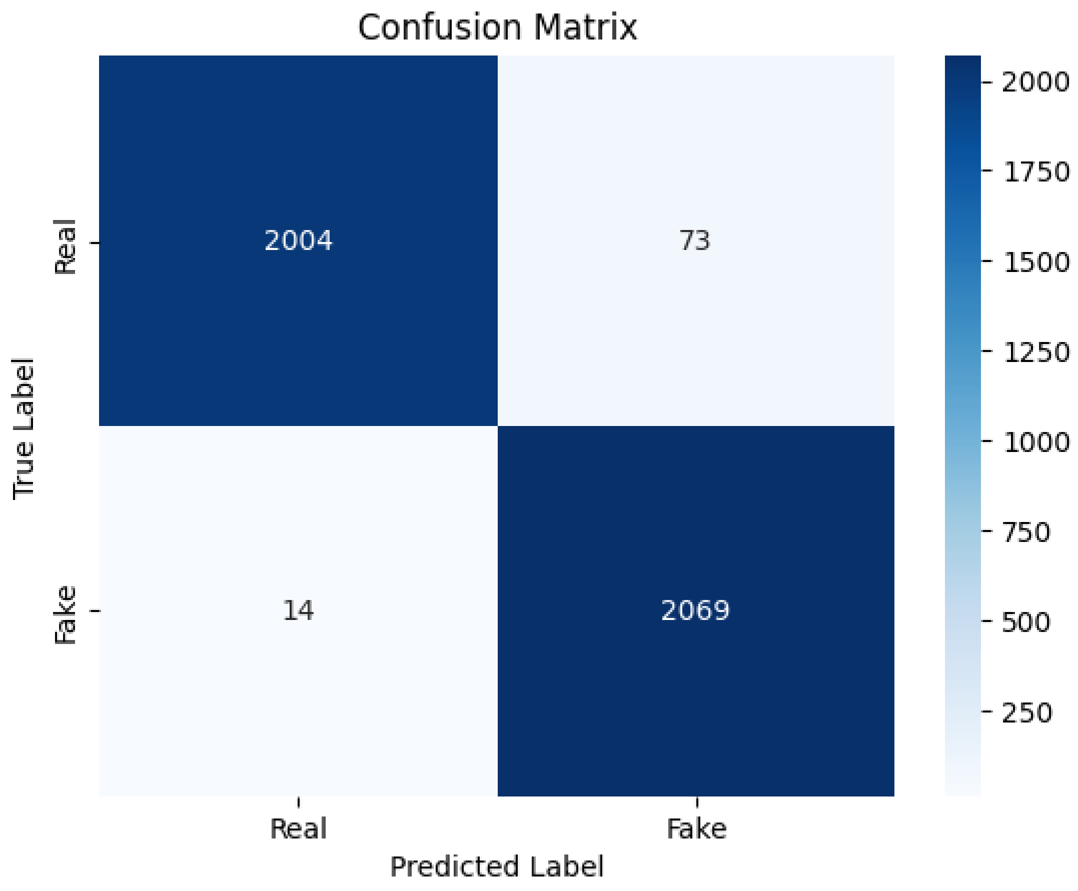


Figure 4

Confusion Matrix

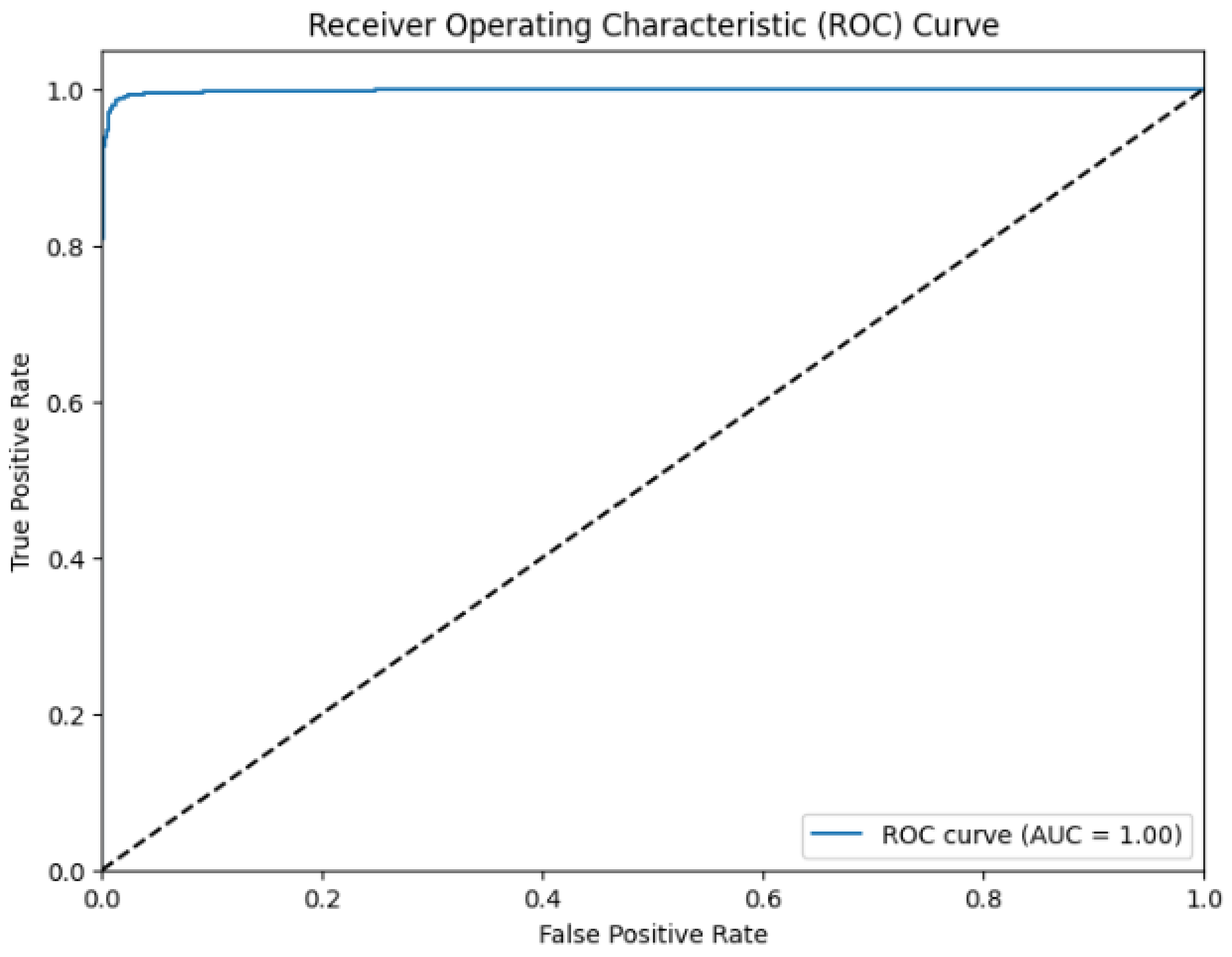


Figure 5

ROC Curve

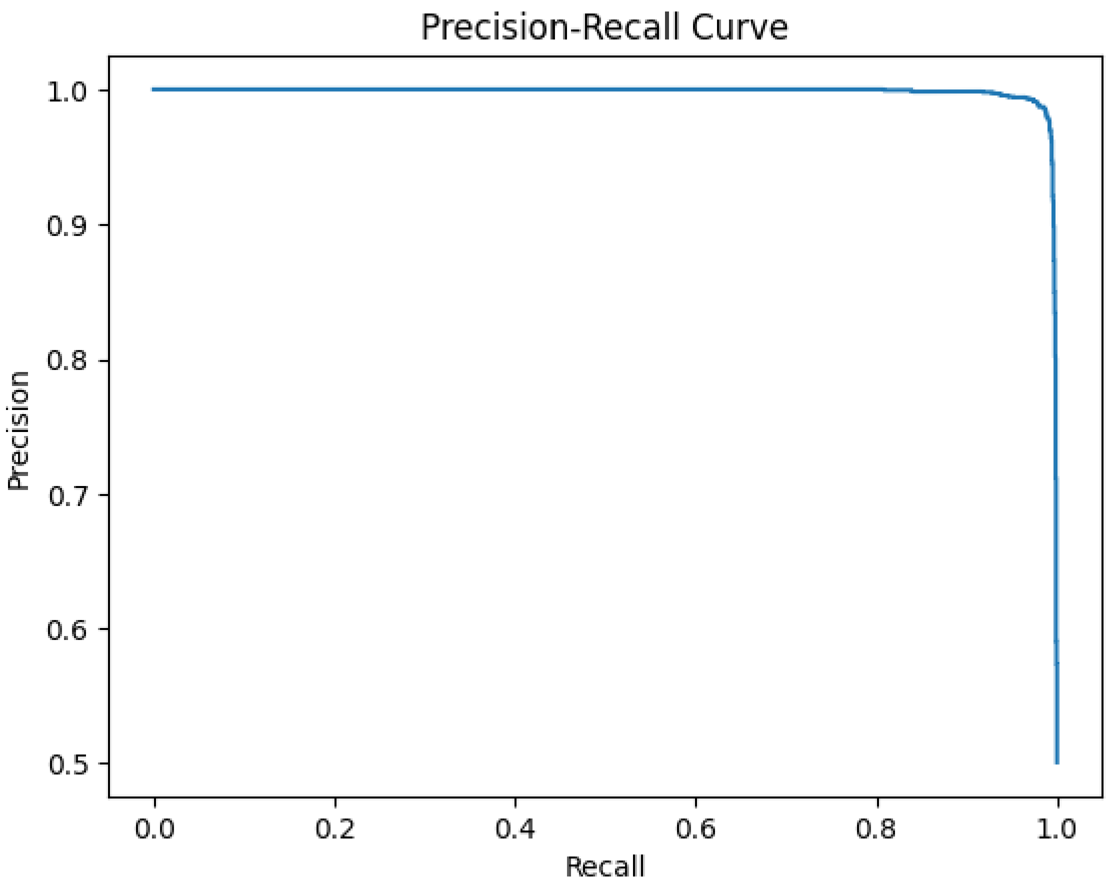


Figure 6

Precision - Recall Curve

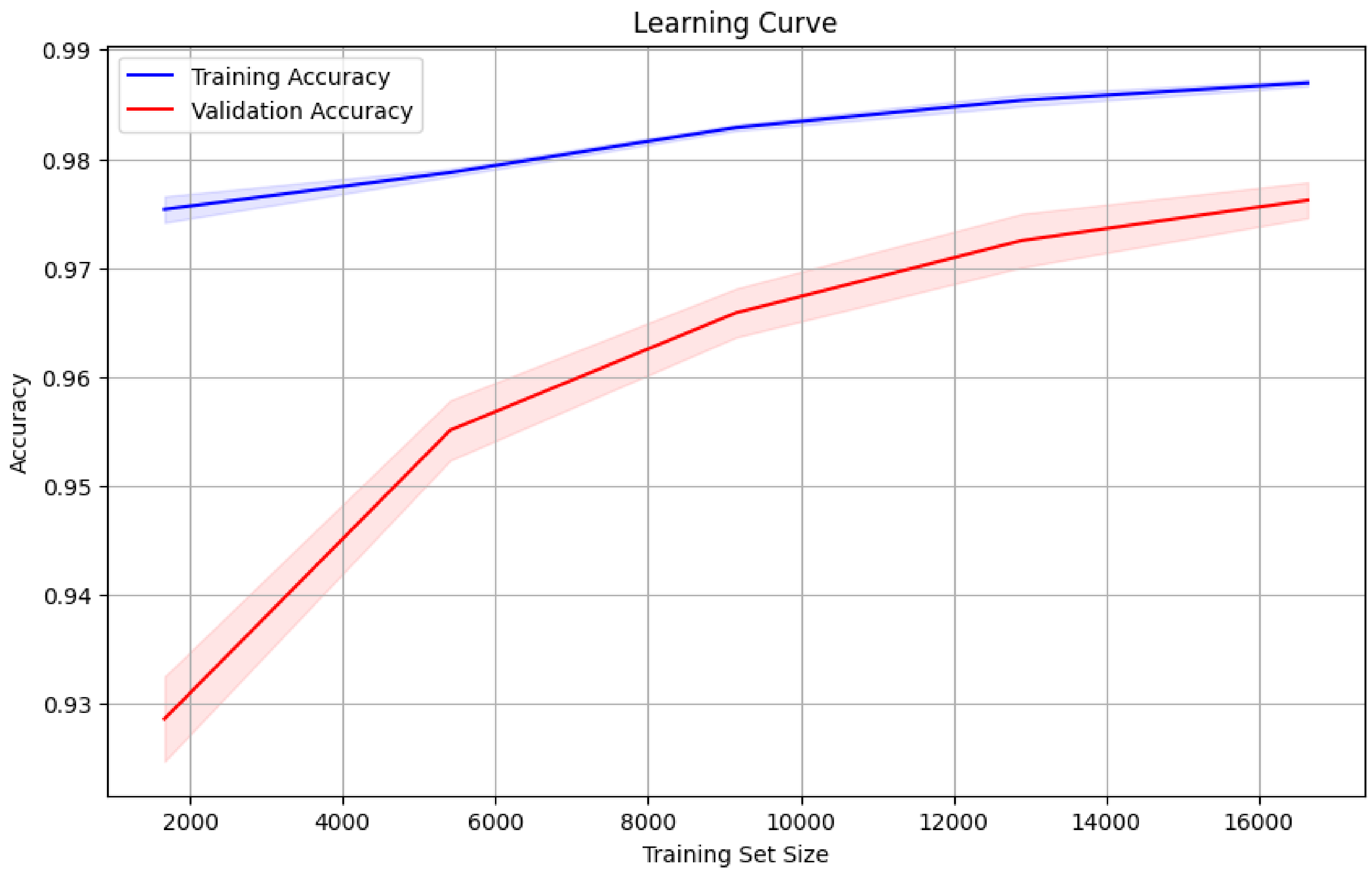


Figure 7

Learning Curve