Advanced Machine Learning for Fake News Detection: Classifiers and Neural Networks

Report of Final Project by Martina Oravcova

Table of Contents

1. Introduction

Motivation
In our digital age, misinformation spreads rapidly, impacting public opinion and democracy. Effective detection tools are essential for maintaining information integrity. This project aims to equip media platforms, OSINT practitioners and the public with reliable tool to distinguish truth from falsehood.

Goal
I am exploring an advanced approach to detecting fake news using traditional classifiers and neural networks to enhance accuracy. The project is based on the CM3060 Natural Language Processing template for Fake News Detection from the University of London. I will compare existing state-of-the-art methods and attempt to match or overcome their results.

Benefits
Offers users such as media platforms, OSINT practitioners, and the general public an advanced tool for quick and accurate fake news detection. Falls under broader categories of media integrity, public information accuracy, and digital information analysis. The specific domain of fake news detection is trust seeking and increasing credibility. The project integrates traditional and advanced methods, which makes it adaptable to new misinformation challenges.

Justification
In today's digital world, the rapid spread of fake news is a growing concern, which impacts public trust and is influencing decisions at both individual and societal levels. Fake news can shape public opinion, manipulate political outcomes, and even cause real-world harm by spreading false information about health, safety, and security. This project addresses these issues and develops an advanced tool for detecting fake news. This is important for maintaining the integrity of information on media platforms. The need for reliable detection methods has become more urgent as misinformation can quickly spread across social networks, create confusion and mislead the public. Because of this, effective solutions for fake news detection are needed for various users, including media outlets, OSINT practitioners, and the general public.

User Needs and Domain Requirements: Media platforms require reliable tools to filter out fake news to maintain their credibility and trustworthiness. OSINT practitioners need accurate and efficient methods to quickly verify information in their intelligence-gathering processes. The general public needs accessible tools that empower them to distinguish between real and fake news to make informed decisions. This project addresses these needs by providing a tool that combines traditional classifiers and neural networks, which allows both interpretability and high accuracy. Traditional classifiers, like Logistic Regression and Support Vector Machines (SVMs), are known for their simplicity and straightforwardness. They are suitable for quick assessments. In contrast, neural networks, such as Convolutional Neural Networks (CNNs), are more adept at capturing complex patterns in textual data. This makes them well-suited for nuanced and sophisticated fake news detection. When I integrate both approaches, the project will offer a comprehensive solution that caters to the varying needs of different user groups.

Innovative Approach and Dataset Choice: The choice of the FakeNewsNet dataset further strengthens the justification for this project. This dataset is widely recognized in the research community for its comprehensive coverage of news content, social context, and temporal information. It has been used in multiple studies mentioned later in the project. The dataset's diverse features—ranging from textual data to social interactions—make it ideal for developing models that can operate effectively in real-world scenarios. When selecting this dataset, the project will for sure contain models trained on a varied set of data. This will allow them to generalize well to different types of fake news. This project also plans to incorporate recent advancements in machine learning and natural language processing and aims to improve upon the results of existing state-of-the-art methods. This innovative approach provides a more effective tool for detecting misinformation, adjusted to the evolving landscape of digital media.

Practical Application and Accessibility: In addition to the technical aspects, the project also includes the development of a user-friendly inference notebook, which is designed to make the detection models accessible to a broader audience. This creates a bridge between complex machine-learning models and end-users who may not have technical expertise. By focusing on usability and accessibility, the project outcomes are not just theoretical but also practical and ready for deployment in real-world settings. The tool is designed with non-technical users in mind, and it can be easily integrated into existing workflows and used effectively by media platforms and OSINT practitioners. User feedback and testing are planned to further refine the tool. This way the project will meet the specific needs of its intended audience.

Conclusion: The project is strongly justified based on a detailed analysis of the domain and user needs. It responds to the requirement for effective fake news detection tools by combining well-established methods with innovative approaches. The careful selection of models and datasets, together with a focus on user accessibility and practical application makes this project highly relevant and beneficial for a wide range of users. It offers a solution that enhances the accuracy and reliability of information in digital media. My project directly addresses the needs of media platforms, OSINT practitioners, and the public in combating misinformation.

2. Literature review

2.1 Fake news during the COVID-19 pandemic

Sudhakar and Kaliyamurthie [1] explore fake news related to COVID-19 misinformation in their study published in *Measurement: Sensors*. The research investigates the effectiveness of various classifiers, including “Naive Bayes, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest and K-Nearest Neighbor. Deep learning models include Convolutional Neural Networks and Long Short-Term Memory (LSTM).” [1], using a large Twitter dataset. The logistic regression and SVM outputted the best results of 95% and 98% accuracy, comparing confusion matrices and ROC of all methods.

The limitation of the study is handling high-volume data (especially manual labelling), which was solved by preprocessing. The authors mention “we collected the actual tweets and then applied the preprocessing of the data we collected” [1], they also mention “We used normalization to manipulate the retrieved raw data.” [1], but don't explain how exactly they did this and I was unable to find the code. Only the generalised schema picture was provided. The paper acknowledges challenges with high-volume data handling and manual labelling but does not explore these limitations deeply. More discussion is needed on how this might limit the scalability and general applicability of the proposed models. I think the accuracy results are impressive and highly dependent on the dataset and its processing which is not explained in detail. In this study, the classification had better accuracy than the neural network. The authors correctly justified the usage of LSTMS as the most suitable neural network for large datasets based on previously published works. The discussion lacks a comparative analysis with existing studies or benchmarks in fake news detection. Without this context, it is difficult to assess the progress or innovation introduced by the study. The study could improve by exploring unsupervised learning methods to reduce reliance on extensive labelled datasets. This process would also help in adapting the models to new and diverse data more efficiently. The authors might consider exploring unsupervised learning methods to mitigate the heavy reliance on labelled datasets.

I will use a different, generalised news dataset than the one used in this study. However, I will apply the same traditional classifiers and evaluation by the confusion matrix and ROC.

2.2 Deep learning and fake news

In their survey published in AI Open, Hu et al. [2] categorise the approaches into supervised, weakly supervised, and unsupervised methods. The survey evaluates different methods based on features like news content, social context, and external knowledge. The techniques are organized into different learning paradigms, which helps to clarify the landscape of FND (Fake News Detection) research. The authors mention, "For each line, we systematically survey the representative methods utilising different features" [2] which indicates a thorough analytical approach. This study assesses existing datasets used for fake news detection. They state, "We introduce several commonly used FND datasets and give a quantitative analysis of the performance of the DL based FND methods over these datasets." [2]. During the dataset analysis, FakeNewsNet caught my interest as one of the best general (non-COVID related) datasets containing text, visual, user profile, repost &

response, network, spatial and temporal data with two labels. The study details that “The dataset contains a total of 23,196 news articles and 69,733 retweets.” [2] The FakeNewsNet performance of specific models in their original published papers was documented in the study. Specifically these methods and their accuracy results: Recursive Neural Network RvNN(ACL18) introduced by Ma et al. [3] with an accuracy of 0.828 on the FakeNewsNet dataset, documented by Song et al. [4] and cited in Hu et al. [2]. Binary Graph Convolutional Network (Bi-GCN) (AAAI20): Developed by Bian et al. [5] with an accuracy of 0.889 on the FakeNewsNet dataset, results documented by Song et al. [4] and discussed in Hu et al. [2]. Temporally Evolving Graph Neural Network for Fake News Detection (TGNF) (IPM21): Presented by Song et al. [4] with an accuracy of 0.935 on the FakeNewsNet dataset, as noted in Hu et al. [2]. Hu et al. confirmed the superior performance of the TGNF model, stating that "On the Ma-Weibo, Ma-Twitter, and FakeNewsNet datasets, TGNF outperforms Bi-GCN, demonstrating the significance of temporal propagation information in detecting the truthfulness of the news." [2]

Shortcomings of the paper emerge in the depth of technique evaluation. The methods and datasets were named and described, but the depth was missing. The method names were abbreviated, and only specialist in the field could understand their meaning. The details and the full names of the methods had to be found in the original papers the survey reviewed, however, the calculated results for the FakeNewsNet dataset of three selected approaches were found in the survey, which has drawn data from Song et al.`s paper [4]. The survey could be enhanced by including case studies or specific instances of real-world applications of these techniques and discussing the ethical implications.

I intend to work with the FakeNewsNet dataset, process it and apply the neural networks approach besides the traditional classifiers, so this paper and its results are a useful baseline, as shown in Table 1.

2.3 Fake news detection dataset

Shen et al. [6] provide a review of methods and challenges in identifying fake news on social media platforms in their survey, "Fake News Detection on Social Networks: A Survey". They explore detection approaches, including content-based, propagation-based, and source-based methods and survey several datasets. Besides others, the FakeNewsNet dataset, which includes text and image content. This confirms that the dataset is important in training fake news detection models.

The study lists numerous approaches and their theoretical base but does not deeply analyse the real-world applicability and success rates of these methods outside controlled environments. I would ask the authors if any of the reviewed datasets are used in a real-world production application. Discussions about the scalability of these detection technologies in diverse and real-time settings are not examined well. My further questions would be if deployed in production in what way the dataset would be updated and how would this be performed in combination with a model retraining with newer data? Authors could critically examine how these varying definitions impact the design and effectiveness of detection algorithms across different platforms and cultural contexts.

My choice of the FakeNewsNet dataset is ideal for understanding and training models to detect misinformation patterns effectively, as it contains a textual, visual, social and spatiotemporal context. This dataset was selected and highlighted by several similar studies including Shen et al. [6], Hu et al. [2], and Shu et al. [7] which justifies my decision for dataset choice.

2.4 FakeNewsNet dataset

The FakeNewsNet dataset, detailed by Shu et al. [7] in their paper "FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media," serves as a resource for research in the detection and analysis of fake news. The dataset is hosted in a GitHub repository FakeNewsNet [8] and contains political and entertainment news, social context and spatiotemporal data. Shu et al. created PolitiFact and GossipCop datasets with true and false news, performed an exploratory analysis of the data, and measured social context and spatiotemporal information. Authors apply "standard machine learning models including support vector machines (SVM), logistic regression (LR), Naive Bayes (NB), and CNN. For SVM, LR, and NB, we used the default settings provided in the scikit-learn and do not tune parameters. For CNN we use the standard implementation with default setting." [7] The results of accuracy, precision, Recall and F1 score were calculated for all methods in both datasets.

The limitations: The datasets PolitiFact and GossipCop may not fully represent the broader spectrum of fake news sources, especially from less mainstream media or languages other than English. The efficacy of models trained on static datasets can decrease over time as the characteristics of fake news evolve. It would be great to have a strategy and infrastructure set up for regular dataset refreshment and model reevaluation. Additional metrics, such as AUC-ROC, could provide a better understanding of model performance. Authors could have used parameter tuning in classification models to improve results and a variety of specialised neural networks to compare the performance and select the best option.

The practical FakeNewsNet dataset limitation is documented in ongoing discussions among researchers using the GitHub repository maintained by Kai Shu and colleagues (GitHub issue #70, 2021) [8]. The Twitter developer API was changed and the dataset has become unavailable. In response, my study will use the FakeNewsNet dataset CSV files downloaded from the GitHub repository FakeNewsNet [8] and the advanced graph version available through the GNN-FakeNews repository [9], maintained by Dou et al., where Shu, K. also participated. In their work, Dou et al. [10] discuss user preference-aware approaches for fake news detection. They focus on the role of graph neural networks in modelling social media data [10].

The outcomes from Shu et al. [7] will serve as baselines for comparison and improvement in my research, as shown in Table 2. Dou et al. [10] work will be a reference for code and logic.

3. Design

3.1. Environment

I will use JupyterNotebooks in a GoogleCollab environment with access to GPUs for intensive computations, I justify this based on the necessity to use large datasets and implement neural networks. For clarity of infrastructure, I will have separate notebook for data processing, separate notebook for classifier training, and separate notebooks for each neural network. I will save the best versions of the models and upload them to Google Drive. The user-facing inference notebook will load the models and allow users to consume them in a user-friendly interface. See Diagram 1. for infrastructure.

3.2. Language and libraries

I will use Python with the required libraries. This decision is justified as all papers I have reviewed for Fake News detection used Python and its libraries.

3.3. Data acquisition

FakeNewsNet dataset source of CSV data is the GitHub repository FakeNewsNet [8] and GNN-FakeNews repository [9] for the graph version of FakeNewsNet processed into the UPFD dataset [10].

3.4. Classifiers

Logistic Regression (baseline Shu et al. [7]), Support Vector Machine (baseline Shu et al. [7]), Naive Bayes (baseline Shu et al. [7])

3.5. Neural Networks

CNN (baseline Shu et al. [7]), optionally depending on time: RvNN (baseline Ma et al. [3]), Bi-GCN (baseline Bian et al. [5]).

3.6. Metrics

I will measure accuracy, precision, recall, and F1-score to assess and refine models. A confusion matrix and ROC will be created.

3.7. Notebook details

3.7.1. Preprocessing notebook

CSV format processing notebook using Pandas and NumPy for data manipulation and analysis. Additional libraries might be required. Depending on the content, I might be handling missing values, and encoding categorical features if any. Normalizing or standardising numerical features might be also needed for neural network models.

3.7.2. Classifier notebook

I will load data from the preprocessing notebook. Shen et al [6] are using the TF-IDF technique for feature extraction, so I will use this too. I will use the scikit-learn library for classifiers baseline comparison, the same way as Shu et al. [7] used basic, not fine-tuned classifiers. I will then compare the results with Shu et al. [7]. I will randomly split the datasets into five parts and conduct 5-fold cross-validation to obtain robust results, as Bian et al. did [5]. Later, I will attempt to fine-tune them to see the model improvement and compare results again. Classifiers I plan to implement: support vector machines (SVM), logistic regression (LR), and Naive Bayes (NB).

3.7.3. CNN notebook

I will load preprocessed data from the preprocessing notebook. Tensorflow library will be used for neural networks, I will use the same basic approach for CNN as Shu et al. [7], based on the work of Denny Britz repository [11]. I will then compare the results of a baseline and attempt to fine-tune the CNN and improve the results.

3.7.4. Bi-GCN notebook

This will be optional and depend on the time and the progress of my project. The notebook will focus on graph format processing. The PyTorch-Geometric library is recommended by Dou et al. [10] to use the dataset in combination with the User Preference-aware Fake News Detection (UPFD) framework.[10], see Diagram 2. First, I will explore the existing setup in the repository to understand the graph dataset. Then I will run Bi-GCN based on Bian et al. [5] and the code in their repository [13].

I will use the Pytorch library for Bi-GCN (Bi-directional Graph Convolutional Network) as described by Bian et al. [5], drawing the logic from Bian's repository [13] originally trained on a different dataset, which I will apply on the FakeNewsNet dataset. Then I will compare the results mentioned by Song et al. [4] and attempt to improve the results.

3.7.5. RvNN notebook

This will be optional and depend on the time and the progress of my project. I will use Theano and NumPy library to recreate Top-down and Bottom-up RvNN (TD-RvNN and BU-RvNN) based on the work of Ma et al. [3], drawing the logic from Ma's repository [12] originally trained on a different dataset, which I will apply on FakeNewsNet dataset or UPFD dataset, whichever is more suitable. Then I will compare the results mentioned by Song et al. [4] and try to improve the results.

3.7.6. Inference notebook

Considering the users as media platforms, OSINT practitioners, and the general public, the consumption of models has to be user-friendly. I will develop a user-friendly publicly accessible Google Collab inference Jupyter notebook utilising the best-performing fake news detection models. The ipywidgets will be used for a user-friendly inference notebook. Users will not need to have access to any other notebook but the inference notebook. The design will include input for text for the user and preprocessing of input text. The code cells will be collapsed to not distract users and to prevent users from accidental modification. The prediction result will be displayed to the user.

3.8. Work Plan

My work plan is described in Table 3.

3.9. Testing Plan

3.9.1. Data Integrity Tests

Data Loading Test: Ensure that the CSV data from FakeNewsNet is loaded correctly, without any corruption or data loss.

Data Split Test: Confirm that the data is split correctly into training, testing, and validation sets. Ensure that the distribution of labels is consistent across these splits.

3.9.2. Model Functionality Tests

Model Training: Verify that all models are training without errors and that loss decreases over epochs as expected.

Convergence Test: Check that the models converge to a reasonable accuracy on the training set and do not show signs of underfitting or overfitting.

3.9.3. Model Evaluation Tests

Cross-Validation: Implement k-fold cross-validation for the classifiers to ensure that the evaluation is robust and the model performance is stable across different data folds.

Performance Metrics: Use accuracy, precision, recall, and F1-score to assess each model. This will help in comparing the models against the baseline studies mentioned in the literature review.

Error Analysis: For models that perform below expectations, conduct an error analysis to understand the common misclassifications.

3.9.4. Integration and Usability Tests

Inference Testing: Test the inference capabilities of the models in the user-friendly interface notebook. Ensure that the models load correctly and predict accurately on new inputs.

Interface Functionality: Test each IPython Widget (text input, buttons) to ensure they interact correctly and trigger the appropriate actions in the notebook.

User Input Handling: Test the system’s response to various types of user input, including edge cases like very long texts, special characters, and empty inputs.

3.9.5. Performance and Scalability Tests

Speed Test: Measure the response time of each model in the inference notebook. This is critical for user satisfaction, especially if deployed in a real-world scenario.

Load Test: Optionally, simulate multiple simultaneous users to see how the system performs under load. This might be more relevant if expecting high traffic or deploying in a scalable environment.

3.9.6. Documentation and Code Review

Code Review: Ensure that all code is well-commented and follows good coding practices. This is important for maintainability, especially if someone else needs to understand or extend the project later.

Documentation Check: Verify that all parts of the project are well-documented, including data sources, model descriptions, and usage instructions for the inference notebook.

3.9.7. Final Checks

Backup and Recovery: Ensure that all models and critical data are backed up appropriately, especially if using cloud environments like Google Collab.

Compliance Check: If required, ensure that the project complies with relevant data use policies, especially in terms of data privacy.

4. Implementation

The Project folder with read access permission is available to the public and contains all relevant project files, including the large models and .pkl files that are not uploaded in GitHub. Here is my Final Project GitHub repository link: https://github.com/7bcp2a/FakeNewsDetection

4.1. Preprocessing notebook

Notebook steps can be seen in Appendix A: FakeNewsNetPreprocessing. This notebook preprocesses the FakeNewsNet dataset created by Shu et al. [7]. The dataset is hosted in a GitHub repository[8] and contains political and entertainment news. The dataset includes the GossipCop and PolitiFact subsets, which will be explored, cleaned, processed, and saved to Google Drive for use in subsequent machine-learning tasks in the Classifier notebook. As displayed in Diagram 3., the datasets have a class imbalance. There are approximately 3 times more records of real news than fake news in the GossipCop dataset. In the PolitiFact dataset, the number of fake news articles is approximately 30% less than that of real news articles. This challenge will require a strategy to balance the dataset. I analysed text length distribution to explore how the length of news titles varies between classes in Diagrams 4. and 5. This can help in feature engineering and in understanding the stylistic differences that might influence classifier performance. The text length distributions indicate that real news has shorter titles compared to fake news in both datasets. Word Frequency Analysis in Diagram 6. identifies the most prevalent terms in each class, for customization of text preprocessing steps (like stopword removal) and feature extraction methods. The results contain stopwords and punctuation, therefore text in the title column needs preprocessing. I cleaned the data by removing unnecessary columns, which prepared the datasets to include only relevant features (text column) for model training. Since most machine learning algorithms require input data, including labels, to be numeric, I converted the labels (label column) to numeric values. I also checked for missing values in all columns of each dataset to ensure the data's integrity. These cleaning steps ensured compatibility with the machine learning algorithms that will be used in the classifier notebook. The preprocessing function converted the text to lowercase, removed punctuation and non-word characters, and performed stemming and stopword removal. This function was applied to the title column of each dataset. As seen in Diagrams 7.-10., converting text to lowercase and removing punctuation reduced the complexity of the text data. The preprocessing steps have prepared the dataset for the Classifier notebook.

4.2. Classifier notebook

Notebook steps can be seen in Appendix B: FakeNewsNetClassifier. This notebook builds and evaluates machine learning models for detecting fake news using the FakeNewsNet dataset. I will be using three different classifiers: Logistic Regression, Support Vector Machine (SVM), and Naive Bayes. I will also balance the dataset using SMOTE and perform hyperparameter tuning to improve the models' performance. I split the dataset the same way as Shu et al. [7]: "We use 80% of data for training and 20% for testing." Shen et al [6] applied the TF-IDF technique for feature extraction, so I used this too. TF-IDF was applied after splitting the data to avoid data leakage, ensuring that the vectorizer was only fitted on the training data and then applied to the test data. SMOTE is a popular technique, that can be applied to balance an imbalanced dataset as described by Chawla et al.[14]. It generates synthetic examples for the minority class. The method combines minority class examples to create synthetic examples and balances the class distribution. It improves classifier performance on imbalanced datasets. I applied this technique to balance the class imbalance present in the original datasets and checked the class distribution again.

I have created a function that trains and evaluates the classifier and visualized the results. The function was reused and called for all three models of Logistic Regression, Support Vector Machine and Naive Bayes. The evaluation included Accuracy, Precision, Recall, F1-score, ROC AUC, Confusion Matrix, Plot of Confusion Matrix and Plot of ROC Curve for all three models on both datasets. Table 4. displays that my models outperformed the baseline results significantly across all metrics (accuracy, precision, recall, F1-score) for both datasets (GossipCop and PolitiFact). After applying SMOTE to balance the datasets, the models showed a more balanced performance across precision and recall, which resulted in higher F1 scores. I have randomly split the datasets into five parts and conducted 5-fold cross-validation to obtain robust results, as Bian et al.[5] did. Cross-validation is a statistical method used to estimate the performance of machine learning models. As Browne et al.[15] explain: “In its simplest form, the leaving one out at a time method, this involves partitioning a sample of size N into a calibration sample of size N-1 and a validation sample of size 1 and repeating the process N times. An average of the N cross-validation index values is then used.” This method involves splitting the data into several subsets (folds), training the model on some subsets while testing it on the remaining subset, and repeating this process several times. The performance metrics are then averaged over all iterations to provide a more robust evaluation. According to Powers et al.[16], “The F1-score, which is the harmonic mean of precision and recall, is particularly recommended for imbalanced datasets because it provides a balance between false positives and false negatives, thus giving a more comprehensive measure of a model's performance.” For this reason, I have selected the F1 score as the scoring parameter in 5-Fold Cross-Validation, because the datasets are imbalanced. The function performs cross-validation on a given model using specified data, folds, and scoring metrics and prints the individual scores along with their average. The results in Picture 1. were consistent across the folds, which was a good sign that the models were generalizing well and not just memorizing the training data. The average performance metrics (F1-score) were very close to the fold-specific metrics. This consistency suggested that the models performed similarly on unseen data, that was another good sign. My use of grid search and hyperparameter tuning was inspired by the methodology described by Chong et al. [17]. I have set arrays of potentially suitable parameters for each model. I have defined the function to perform hyperparameter tuning and evaluation with 5-fold cross-validation. The function was then called and performed hyperparameter tuning and evaluation for each model. The search was computationally intense and ran for approximately 1h. The output contained the Best Parameters, Best cross-validation score, Accuracy, Precision, Recall, F1-Score, ROC AUC, Confusion Matrix, Plot of Confusion Matrix and Plot of ROC Curve. Hyperparameter tuning has led to some improvements as seen in Table 5., especially in the case of SVM on the GossipCop dataset. However, for Logistic Regression and Naive Bayes, the changes are minimal. I selected svm_model_gossipcop as the best model after hyperparameter tuning, saved it to the file and also saved the TF-IDF vectorizer to a file. You can see the statistics in Picture 2., Diagram 11. and Diagram 12. My models significantly outperform the baseline results across all metrics for both datasets (GossipCop and PolitiFact). Balancing the datasets with SMOTE and tuning hyperparameters further improved the models' performance. Based on the evaluation metrics (accuracy, precision, recall, F1-score, and ROC AUC) for each classifier, the Support Vector Machine with hyperparameter tuning on the GossipCop dataset seems to perform the best overall.

4.3. CNN notebook

Notebook steps can be seen in Appendix C: FakeNewsNetCNN. This notebook builds and evaluates the Convolutional Neural Network (CNN) model for detecting fake news using the FakeNewsNet dataset from the Preprocessing notebook. This work is based on the work of Shu et al. [7] and Denny Britz repository. [11] Functions to set seeds and preserve deterministic operations are defined for reproducibility. Features (X) and labels (y) are defined for both datasets. Text data is tokenized, padded, and split into training and testing sets. Tokenization converts text into numerical values. This process makes it suitable for neural network operations. Padding keeps uniform input length and enables efficient batch processing as explained by Denny Britz's blog post linked to his repository[11]. Encoding labels into numerical form standardizes the output for classification tasks. I am splitting the dataset the same way as Shu et al. [7]: "We use 80% of data for training and 20% for testing." Separate CNN models for GossipCop and PolitiFact are built using embedding, convolutional, pooling, dropout, and dense layers. Models are compiled with Adam optimizer and binary cross-entropy loss. I have experimented with different layers, parameters and settings explained by Denny Britz [11]. Class weights are calculated to handle class imbalance in the datasets. These weights are used during model training to balance the impact of each class. The CNN models are trained with early stopping and learning rate reduction callbacks. Performance metrics accuracy, precision, recall, F1-score, and ROC AUC are calculated and displayed, along with confusion matrices and ROC curves. The F1 score is used during training and evaluation. Keras doesn't natively support the F1 score as a metric during training. I keep accuracy as a metric for monitoring during training because it's fast and gives a general sense of model performance. I use a custom callback to log the F1 score at the end of each epoch. Looking at Diagram 13., F1 score graphs show an increase over the epochs. The loss graphs show a steady decrease in both training and validation loss at the beginning, which stabilizes towards the later epochs. Table 6. Shows how my models significantly outperformed Shu et al.'s CNN results in all metrics for both datasets. I could not use scikeras for k-fold cross-validation due to compatibility issues with existing models, so I implemented k-fold cross-validation manually. 5-Fold Cross-Validation results are in Picture 3. I selected the Keras tuner for my CNN hyperparameter tuning. Keras Tuner demonstrates superior accuracy in CNN applications (see Table II in Halim et al.[18]). In Table 7., for the GossipCop dataset, the tuned model has improved in most metrics, so this is my preferred model. For the PolitiFact dataset, the results are mixed. Because of this, I choose the original model. My models significantly outperform the baseline results across all metrics for both datasets. Calculated class weights in both datasets handle class imbalance. Hyperparameter tuning improved GosspiCop model performance, which was not the case for the PolitiFact model. The best-performing model was GossipCop with tuned hyperparameters. The CNN models for both datasets had better performance than traditional classifiers, so I used them in the inference notebook.

4.4. Inference notebook

Notebook steps can be seen in Appendix D: FakeNewsNetInference. This notebook helps users distinguish between fake and real news titles using CNN models created in the FakeNewsNetCNN notebook. For more information, please see the project folder, and start with the Report.pdf. The code is hidden to create a user-friendly interface, Picture 4.

5. Evaluation

I achieved my goal from the introduction to compare existing state-of-the-art methods for detecting fake news using traditional classifiers and neural networks to enhance accuracy, based on the CM3060 Natural Language Processing template for Fake News Detection from the University of London. I significantly surpassed the baseline results of FakeNewsNet by Shu et al. [7] available in Table 2. Comparison can be seen in Table 4. and Table 6. I fine-tuned the models of traditional classifier and CNN models, which produced generally improved results, as seen in Tables 5. and 7. I completed compulsory Preprocessing, Classifier, CNN and Inference notebooks as planned. I have not completed the optional Bi-GCN and RvNN notebook yet, due to lack of time and work on existing notebooks and the Report. If time allows it, I might be successful in completing also optional notebooks. I am progressing well with the working plan as described in Table 3. The following tests were completed.

5.1. Data Integrity Tests

Data Loading Test:

FakeNewsNetPreprocessing notebook loads raw data and preprocesses it so other notebooks don't have to repeat the process. CNN and Classifier notebooks start with a data loading process where data from the FakeNewsNetPreprocessing is imported. Each dataset was displayed using the head() method to visually confirm the correct loading of data. The integrity of this data is verified by inspecting initial entries and confirming the absence of null values post-load.

Data Split Test:

Data is split into training and testing sets with a typical distribution of 80/20 for training/testing, as outlined by Shu et al [7]. This split is important for unbiased model evaluation. The distribution consistency of labels was not verified, but I used fixed random state 42 in Classifier and CNN notebooks to make sure the splits were reproducible. This means that any imbalances or distribution inconsistencies were consistently handled across different runs. I used SMOTE in Classifier models and class weights in CNN models to handle class imbalance. This indirectly helps in maintaining performance consistency even if the label distribution in the training/testing splits isn't perfect.

5.2. Model Functionality Tests

Model Training:

During the training process, the training loss was monitored for a consistent decrease and model weights were updated. This showed that models learned from the training data.

Convergence Test:

Models are evaluated for convergence by examining training and validation losses. After discovering that datasets are imbalanced, I did not focus on increasing accuracy, but an F1 score metric for evaluation as recommended by Powers et al.[16]. I prevented overfitting by applying early stopping and reduced learning rate in the CNN notebook when loss stopped improving.

5.3. Model Evaluation Tests

Cross-Validation:

The stability and reliability of model performance are verified through k-fold cross-validation for the models. This method measures model effectiveness across different subsets of data, and checks that performance metrics are not dependent on a particular split of data. I used 5-fold cross-validation and the results were consistent in Classifier models and CNN models.

Performance Metrics:

Accuracy, precision, recall, and F1-score are calculated for each model, I also compare confusion matrices and ROC curves in charts. I selected the best-performing models based on metrics comparisons for usage in the inference notebook FakeNewsNetInference.

Error Analysis:

I compared the evaluation results of all models, including the bad ones. I did not use bad models as other models performed better. Before this, I made sure the models were correctly set up and trained. Despite the correct setup, they had bad results in the evaluation. Even in cross-validation, the results were consistently worse and similar after hyperparameter tuning. Such models were Logistic Regression and Naive Bayes.

5.4. Integration and Usability Tests

Inference Testing:

The functionality of CNN models within a user-friendly notebook interface was tested to ensure they load correctly and perform real-time predictions accurately. The best models were loaded and titles of news were submitted outputting the results.

Interface Functionality:

Interactive components like text input, dropdown and buttons were tested across browsers to ensure consistent functionality. This confirmed that end-users can interact with the system as intended without technical difficulties.

User Input Handling:

The system's robustness against various user inputs was also tested. Function validate_input considers cases for empty input, or short input (less than three words).

5.5. Performance and Scalability Tests

Speed Test:

The model's response time was measured in the inference notebook to confirm that the prediction latency is within acceptable limits for real-time use. The progress bar of inference shows the processing to the user and is within reasonable limits (~2 seconds). Fast response times are the priority for user satisfaction and usability. Connecting to Collab runtime took approximately 9 seconds. Running the inference code to load the models and set up the UI took approximately 36 seconds. This time was longer, but was run only at the beginning, after that the UI was active and the user could keep checking the news. In case of a need to run inference continuously, a different, probably paid setup should be considered.

Load Test:

This optional test was not performed, as my project is mostly theoretical and inference notebook is just a simple interface demonstrating the ability of models in a practical manner. The free version of Google Collab hosts the notebook and has limitations on simultaneous sessions and computational resources, which makes extensive load testing impractical. My current setup is sufficient for the current scope that focuses on model functionality rather than scalability in a production environment.

5.6. Documentation and Code Review

Code Review:

Code across notebooks was reviewed for clarity and maintainability. I have added relevant comments and adhered to coding standards. This will make future modifications or extensions easier.

Documentation Check:

I have iteratively created a documentation Report.pdf in the project folder, that describes the project and its components. The inference notebook contains clear and user-friendly instructions on how to use it right after the description of the notebook.

5.7. Final Checks

Backup and Recovery:

The Project folder with read access permission is available to the public in the Cloud Environment of Google Collab and contains all relevant project files. Backups of all these files are also kept locally and in private Google Drive folders to prevent data loss and potential system failures.

Compliance Check:

All data handling processes are reviewed to confirm compliance with relevant data privacy laws and ethical guidelines, safeguarding user data and maintaining trust. The dataset that was used for model training is publicly available with the MIT license. All authors, sources, used methods and code have been credited and mentioned in the References of the Report and notebooks. Users running FakeNewsNetInference in Google Collab log in and submit the input text of news titles from their own free decision and will and their data is not saved anywhere.

6. Conclusion

My project introduced originality and enhanced fake news detection by combining traditional classifiers and neural networks. Based on the CM3060 Natural Language Processing template for Fake News Detection from the University of London, the developed methods were tested through a series of notebooks that covered different stages of the machine learning workflow including data preprocessing, model training, evaluation, and deployment.

Achievements:

1. Data Preparation and Processing
- The FakeNewsNet dataset was prepared through preprocessing to prepare data and to make sure the data was ready for machine learning application and suitable for modelling.

2. Model Implementation and Evaluation
- The project implemented and evaluated several machine learning models such as Logistic Regression, Support Vector Machine, Naive Bayes, and Convolutional Neural Networks (CNNs). These models were tuned and tested using accuracy, precision, recall, F1-score, Confusion Matrix and ROC AUC with results better than the baseline.
- CNNs showed better performance compared to traditional classifiers and were used in the Inference notebook for practical usage to demonstrate real-world application of my project.

3. Testing and Validation
- Testing such as data integrity checks, model functionality, and performance evaluations confirmed the models' reliability.
- Cross-validation results showed that the models could generalize well across different data subsets.
- The inference notebook allowed the practical demonstration of the models in real-time and proved their practical potential.

4. Documentation and Usability
- The project included detailed documentation and code reviews so that future researchers or practitioners could easily understand and build on this work.

Impact:

The project's results enhance the tools available for the detection of fake news and provide accurate and efficient models useful for media platforms, OSINT practitioners, and the general public.

Limitations:
The most significant limitation is that the performance of my models is limited and depends on the specific characteristics of the FakeNewsNet dataset, which might not include certain types of misinformation or represent less common forms of fake news. The GossipCop dataset subset was much smaller and the results were not as precise as in the PolitiFact subset, this led to an overfitting tendency that had to be addressed. Both subsets had class imbalances and this issue had to be handled, which might have had an impact on the results. The FakeNewsNet dataset is not new, it was updated in 2019 and therefore models' capability is affected by this. Ideally, the dataset would be regularly updated and models retrained and reevaluated. For this, a machine learning pipeline would be beneficial. Another option would be online learning mechanisms that allow models to adapt to new types of fake news as they emerge without needing full retraining, such as an adaptive learning system. The FakeNewsNet dataset is in English, so the models are limited by regions and can not be used in locations where other languages are predominant. Cultural bias must be mentioned, as these locations can also have different interests in terms of gossip or politics than those present in this dataset.

Future Work:
The immediate future work could add the optional notebooks. Long-termly, I could look at including more diverse datasets including non-English content and media from a wider range of sources to increase the models' applicability for global use, for cross-language and cross-cultural adaptation. This could require potential collaborations with linguists and cultural experts to enhance understanding and detection across different regions. Exploring unsupervised and semi-supervised learning could also reduce reliance on labelled datasets, which are often a bottleneck in scaling detection systems. The newest trends tend to utilise large language models for fake news detection and this use case can be also explored, like transformer models. Potential future deployment strategies could include integration with social media platforms or news aggregators to access real-time data and feedback. This would require robust, probably cloud-based infrastructure and high computational costs for continuous training setup. Partnerships with tech companies for real-world application tests can be also considered, especially in OSINT.

Ethical Considerations:
There is a potential for bias, due to the limitations of dataset timing, language, culture, subsets and sources that were used, risk of news censorship in the used dataset and potentially new future datasets. AI-generated news, either fake or not must be also mentioned which will add another dimension into consideration. Fake news detection tools are never 100%-ly correct and can output false positives and false negatives. This must be considered as errors could affect public discourse or individual reputations. We have to keep in mind the risk of misuse of the technology.

This project met its goals, proved that it is possible to improve and overcome published scientific results and created a strong basis for further research in fake news detection. The developed models are ready to be used and have the potential to improve the integrity of information in digital media. Through ongoing improvement and expansion, these models can combat misinformation.

7. References

[1] SUDHAKAR, M. and KALIYAMURTHIE, K.P. 2024. Detection of fake news from social media using support vector machine learning algorithms. Measurement: Sensors, 32, 101028. https://doi.org/10.1016/j.measen.2024.101028

[2] HU, L., LIU, Q., GUO, N., YUAN, J., and YANG, Y. 2023. Deep learning for fake news detection: A comprehensive survey. AI Open, 3, 133-155. https://doi.org/10.1016/j.aiopen.2022.09.001

[3] MA, J., GAO, W., and WONG, K.-F. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1980–1989, Melbourne, Australia, July. Association for Computational Linguistics. DOI: 10.18653/v1/P18-1184

[4] SONG, C., SHU, K., WU, B., et al. 2021. Temporally Evolving Graph Neural Network for Fake News Detection. Information Processing and Management, 58, 102712. https://doi.org/10.1016/j.ipm.2021.102712

[5] BIAN, T., XIAO, X., XU, T., ZHAO, P., HUANG, W., RONG, Y., and HUANG, J. 2020. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. arXiv preprint arXiv:2001.06362. https://doi.org/10.48550/arXiv.2001.06362

[6] SHEN, Y., LIU, Q., GUO, N., YUAN, J., YANG, Y. 2023. Fake News Detection on Social Networks: A Survey. Applied Sciences, 13, 11877. https://doi.org/10.3390/app132111877

[7] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. https://doi.org/10.48550/arXiv.1809.01286

[8] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context, and Dynamic Information for Studying Fake News on Social Media. https://github.com/KaiDMML/FakeNewsNet

[9] DOU, Y., SHU, K., XIA, C., YU, P.S., and SUN, L. 2021. Dataset for User Preference-aware Fake News Detection. https://github.com/safe-graph/GNN-FakeNews

[10] DOU, Y., SHU, K., XIA, C., YU, P.S., and SUN, L. 2021. User Preference-aware Fake News Detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 11–15, 2021, Virtual Event, Canada. arXiv:2104.12259. https://doi.org/10.48550/arXiv.2104.12259

[11] BRITZ, D. 2021. Convolutional Neural Network for Text Classification in TensorFlow. https://github.com/dennybritz/cnn-text-classification-tf

[12] MA, J. 2018. Rumor_RvNN. https://github.com/majingCUHK/Rumor_RvNN

[13] BIAN, T., XIAO, X., and XU, T. 2020. Bi-Directional Graph Convolutional Networks for Fake News Detection on Social Media. https://github.com/TianBian95/BiGCN

[14] CHAWLA, N.V., BOWYER, K.W., HALL, L.O., and KEGELMEYER, W.P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.48550/arXiv.1106.1813

[15] BROWNE, M.W. 2000. Cross-Validation Methods. Journal of Mathematical Psychology, 44, 108–132. https://doi.org/10.1006/jmps.1999.1279

[16] POWERS, D.M.W. 2011. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. International Journal of Machine Learning Technology, 2(1), pp. 37-63. https://doi.org/10.48550/arXiv.2010.16061

[17] CHONG, K. and SHAH, N. 2022. Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis. International Journal of Advanced Computer Science and Applications, 13(12), pp. 90-94. DOI: 10.14569/IJACSA.2022.0131213

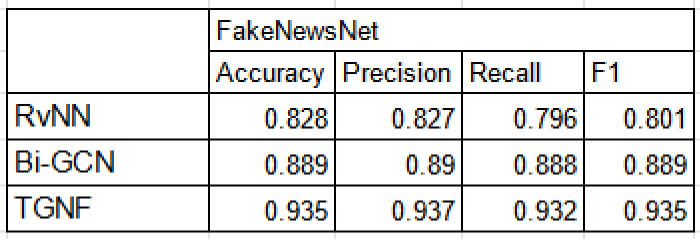
[18] HALIM, A., CHOW, C., BUDIHARTO, M., ACHMAD, S., and SUTOYO, R. 2023. The Impact of Hyperparameter Tuning in Convolutional Neural Network on Image Classification Model: A Case Study of Plant Disease Detection. In Proceedings of the IEEE Conference on Innovative Research and Development (ICORIS), pp. 1-6. https://doi.org/10.1109/ICORIS60118.2023.10352209

[19] PYTORCH GEOMETRIC TEAM. 2021. UPFD Example Implementation in PyTorch Geometric. https://github.com/pyg-team/pytorch_geometric/blob/master/examples/upfd.py

[20] FEY, Matthias and LENSSEN, Jan E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In Proceedings of the ICLR Workshop on Representation Learning on Graphs and Manifolds. https://doi.org/10.48550/arXiv.1903.02428

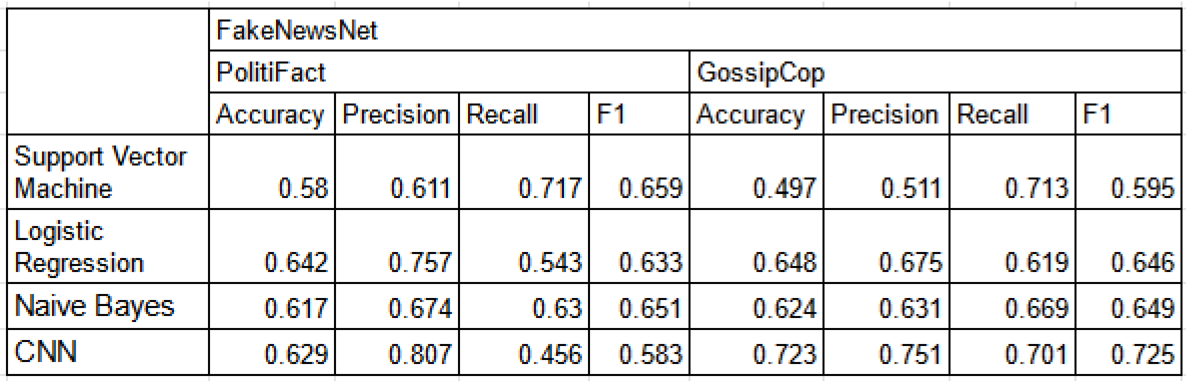
8. Appendices

Table 1. Results of FakeNewsNet from Song et al. [4]



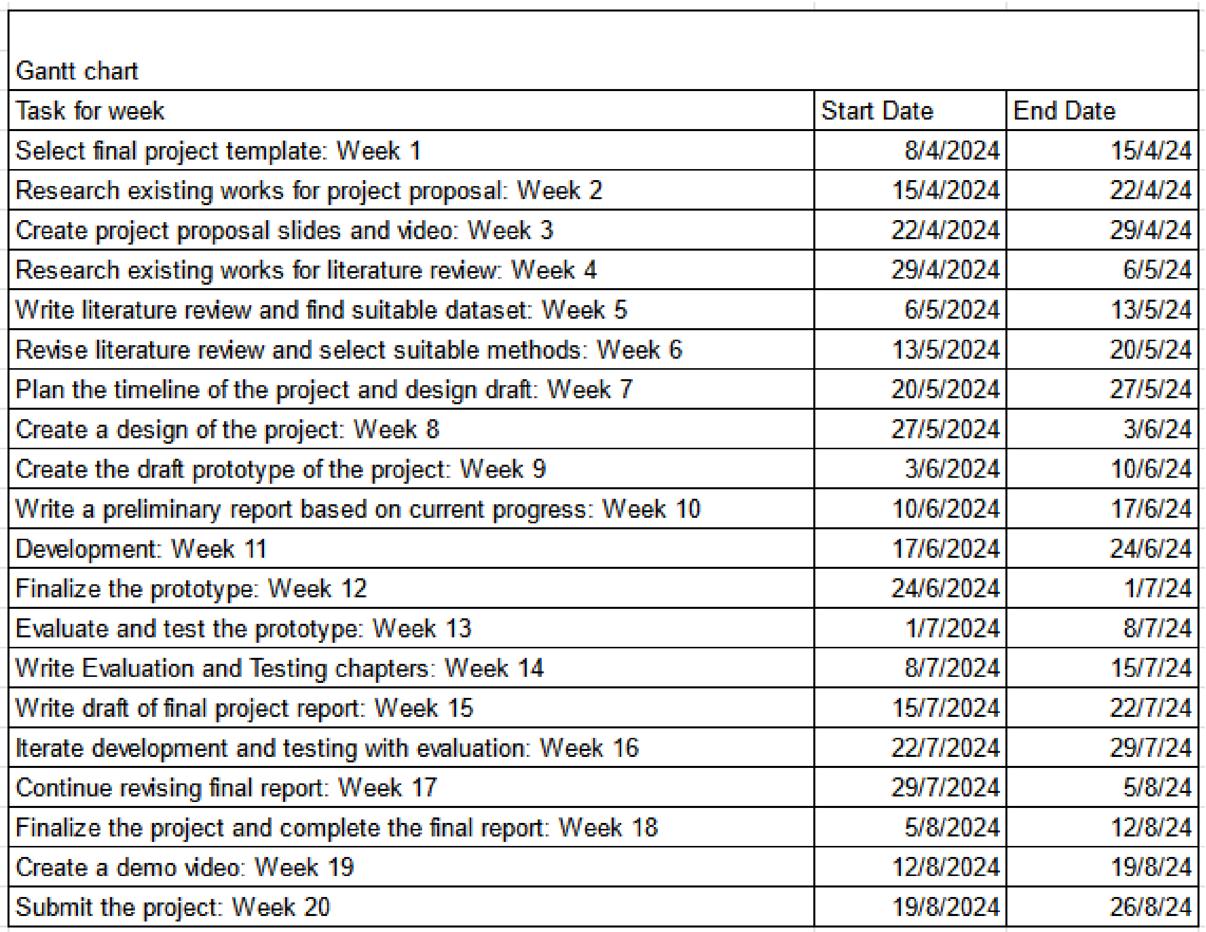
	FakeNewsNet			
	Accuracy	Precision	Recall	F1
RvNN	0.828	0.827	0.796	0.801
Bi-GCN	0.889	0.89	0.888	0.889
TGNF	0.935	0.937	0.932	0.935

Table 2. Results of FakeNewsNet from Shu et al. [7]



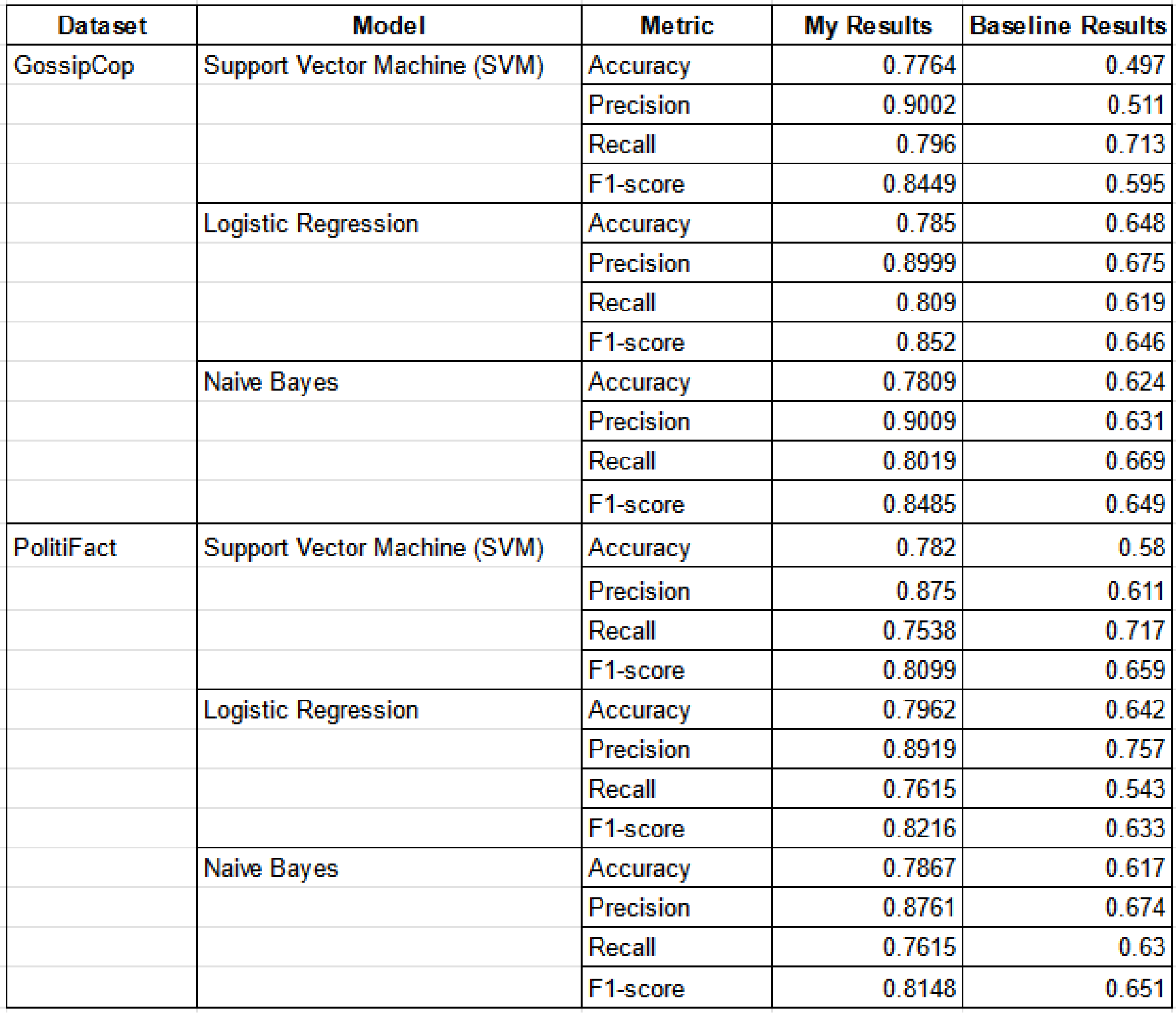
	FakeNewsNet							
	PolitiFact				GossipCop			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Support Vector Machine	0.58	0.611	0.717	0.659	0.497	0.511	0.713	0.595
Logistic Regression	0.642	0.757	0.543	0.633	0.648	0.675	0.619	0.646
Naive Bayes	0.617	0.674	0.63	0.651	0.624	0.631	0.669	0.649
CNN	0.629	0.807	0.456	0.583	0.723	0.751	0.701	0.725

Table 3. Gantt Chart



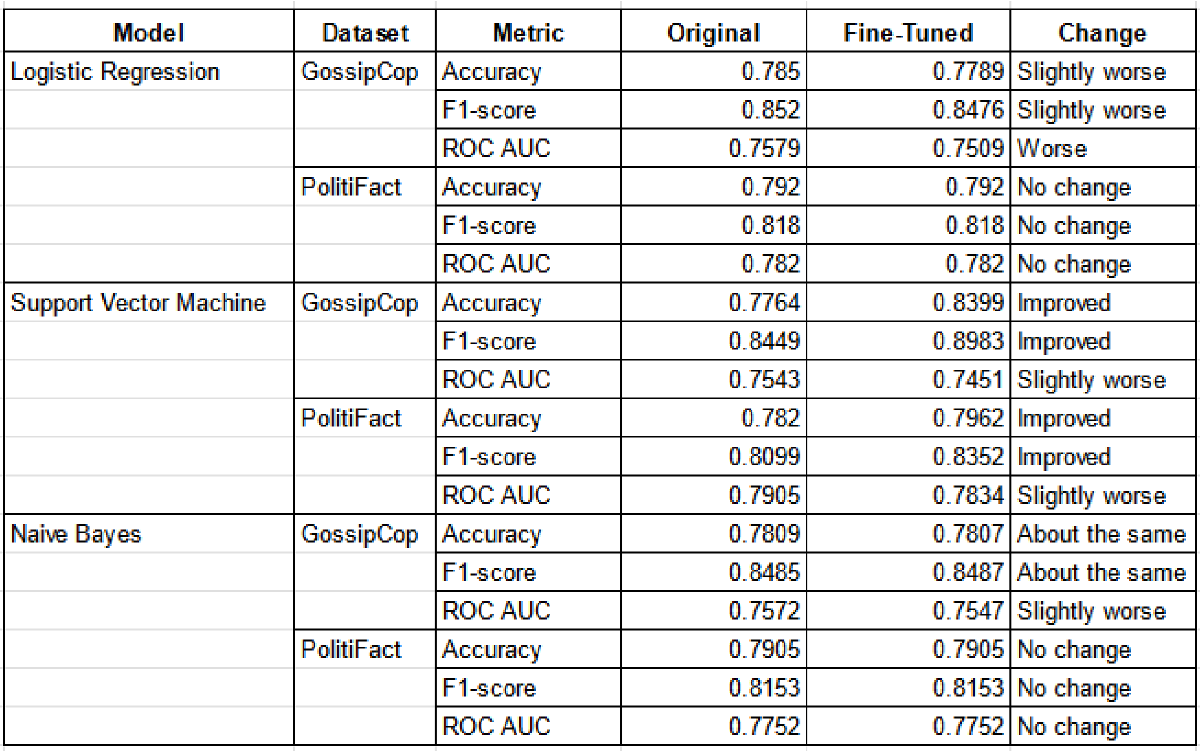
Gantt chart		
Task for week	Start Date	End Date
Select final project template: Week 1	8/4/2024	15/4/24
Research existing works for project proposal: Week 2	15/4/2024	22/4/24
Create project proposal slides and video: Week 3	22/4/2024	29/4/24
Research existing works for literature review: Week 4	29/4/2024	6/5/24
Write literature review and find suitable dataset: Week 5	6/5/2024	13/5/24
Revise literature review and select suitable methods: Week 6	13/5/2024	20/5/24
Plan the timeline of the project and design draft: Week 7	20/5/2024	27/5/24
Create a design of the project: Week 8	27/5/2024	3/6/24
Create the draft prototype of the project: Week 9	3/6/2024	10/6/24
Write a preliminary report based on current progress: Week 10	10/6/2024	17/6/24
Development: Week 11	17/6/2024	24/6/24
Finalize the prototype: Week 12	24/6/2024	1/7/24
Evaluate and test the prototype: Week 13	1/7/2024	8/7/24
Write Evaluation and Testing chapters: Week 14	8/7/2024	15/7/24
Write draft of final project report: Week 15	15/7/2024	22/7/24
Iterate development and testing with evaluation: Week 16	22/7/2024	29/7/24
Continue revising final report: Week 17	29/7/2024	5/8/24
Finalize the project and complete the final report: Week 18	5/8/2024	12/8/24
Create a demo video: Week 19	12/8/2024	19/8/24
Submit the project: Week 20	19/8/2024	26/8/24

Table 4. Comparison of my Classifier results with baseline Shu et al. [7]



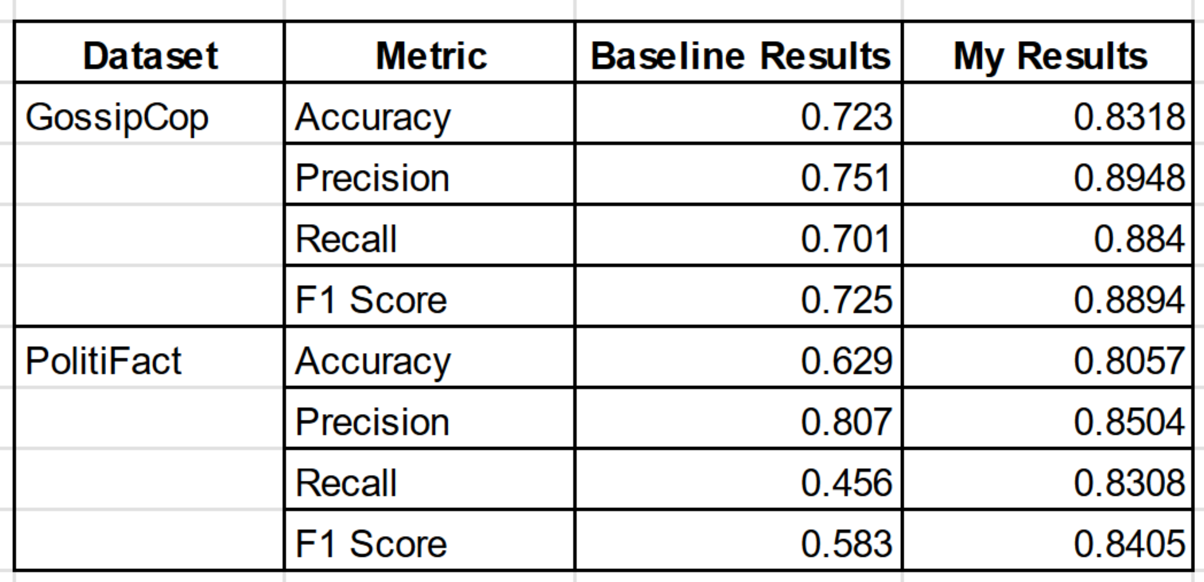
Dataset	Model	Metric	My Results	Baseline Results
GossipCop	Support Vector Machine (SVM)	Accuracy	0.7764	0.497
		Precision	0.9002	0.511
		Recall	0.796	0.713
		F1-score	0.8449	0.595
	Logistic Regression	Accuracy	0.785	0.648
		Precision	0.8999	0.675
		Recall	0.809	0.619
		F1-score	0.852	0.646
	Naive Bayes	Accuracy	0.7809	0.624
		Precision	0.9009	0.631
		Recall	0.8019	0.669
		F1-score	0.8485	0.649
PolitiFact	Support Vector Machine (SVM)	Accuracy	0.782	0.58
		Precision	0.875	0.611
		Recall	0.7538	0.717
		F1-score	0.8099	0.659
	Logistic Regression	Accuracy	0.7962	0.642
		Precision	0.8919	0.757
		Recall	0.7615	0.543
		F1-score	0.8216	0.633
	Naive Bayes	Accuracy	0.7867	0.617
		Precision	0.8761	0.674
		Recall	0.7615	0.63
		F1-score	0.8148	0.651

Table 5. Comparison of metrics in Classifier models before and after Hyperparameter tuning



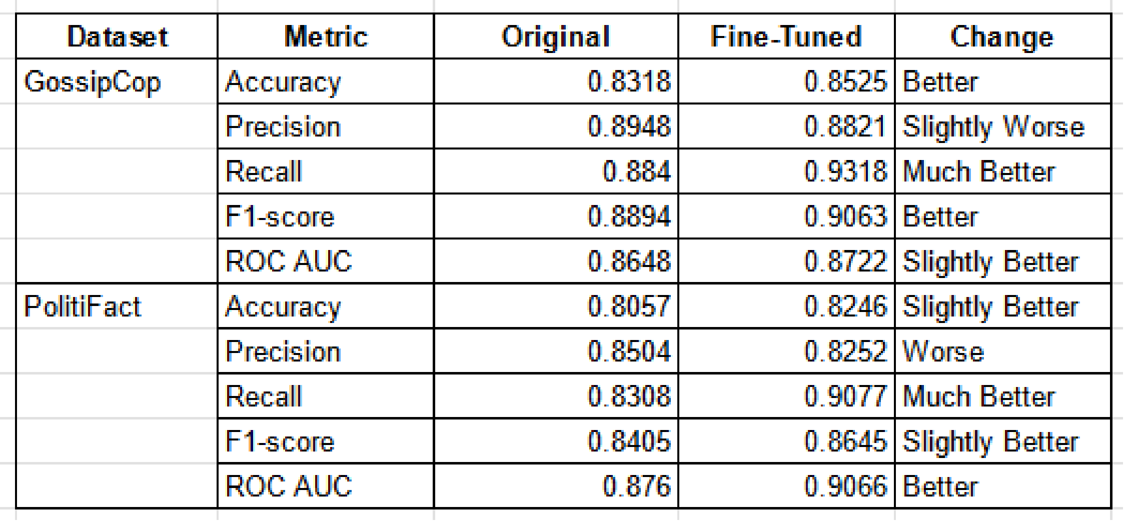
Model	Dataset	Metric	Original	Fine-Tuned	Change
Logistic Regression	GossipCop	Accuracy	0.785	0.7789	Slightly worse
		F1-score	0.852	0.8476	Slightly worse
		ROC AUC	0.7579	0.7509	Worse
	PolitiFact	Accuracy	0.792	0.792	No change
		F1-score	0.818	0.818	No change
		ROC AUC	0.782	0.782	No change
Support Vector Machine	GossipCop	Accuracy	0.7764	0.8399	Improved
		F1-score	0.8449	0.8983	Improved
		ROC AUC	0.7543	0.7451	Slightly worse
	PolitiFact	Accuracy	0.782	0.7962	Improved
		F1-score	0.8099	0.8352	Improved
		ROC AUC	0.7905	0.7834	Slightly worse
Naive Bayes	GossipCop	Accuracy	0.7809	0.7807	About the same
		F1-score	0.8485	0.8487	About the same
		ROC AUC	0.7572	0.7547	Slightly worse
	PolitiFact	Accuracy	0.7905	0.7905	No change
		F1-score	0.8153	0.8153	No change
		ROC AUC	0.7752	0.7752	No change

Table 6. Comparison of my CNN results with baseline Shu et al. [7]



Dataset	Metric	Baseline Results	My Results
GossipCop	Accuracy	0.723	0.8318
	Precision	0.751	0.8948
	Recall	0.701	0.884
	F1 Score	0.725	0.8894
PolitiFact	Accuracy	0.629	0.8057
	Precision	0.807	0.8504
	Recall	0.456	0.8308
	F1 Score	0.583	0.8405

Table 7. Comparison of metrics in CNN models before and after Hyperparameter tuning



Dataset	Metric	Original	Fine-Tuned	Change
GossipCop	Accuracy	0.8318	0.8525	Better
	Precision	0.8948	0.8821	Slightly Worse
	Recall	0.884	0.9318	Much Better
	F1-score	0.8894	0.9063	Better
	ROC AUC	0.8648	0.8722	Slightly Better
PolitiFact	Accuracy	0.8057	0.8246	Slightly Better
	Precision	0.8504	0.8252	Worse
	Recall	0.8308	0.9077	Much Better
	F1-score	0.8405	0.8645	Slightly Better
	ROC AUC	0.876	0.9066	Better

Diagram 1. Infrastructure

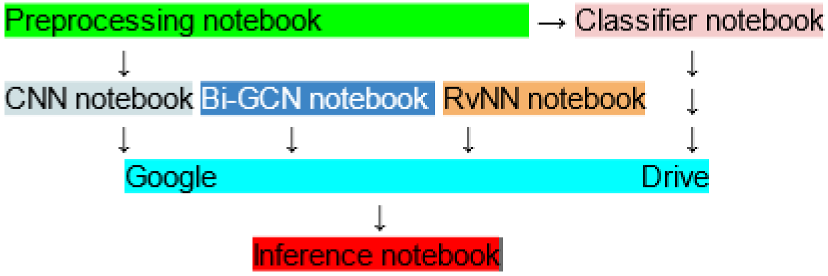


Diagram 2. Data flow chart of UPFD framework by Dou et al. [10]

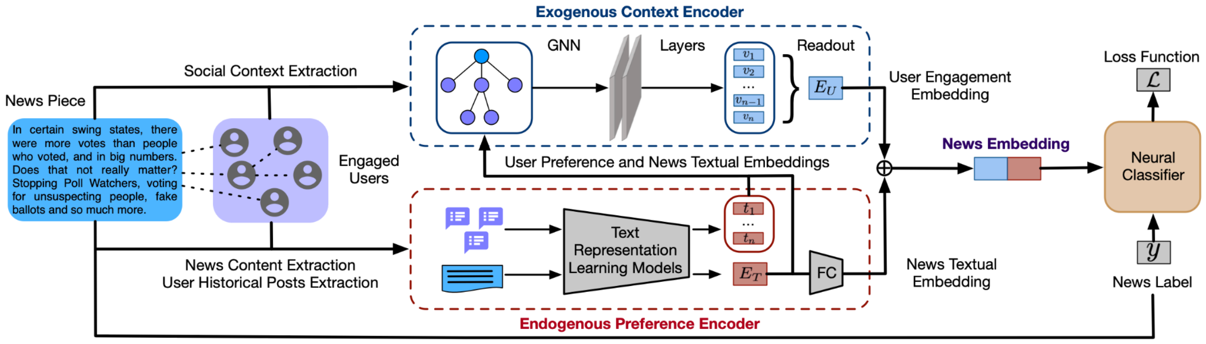


Diagram 3. Class Distribution

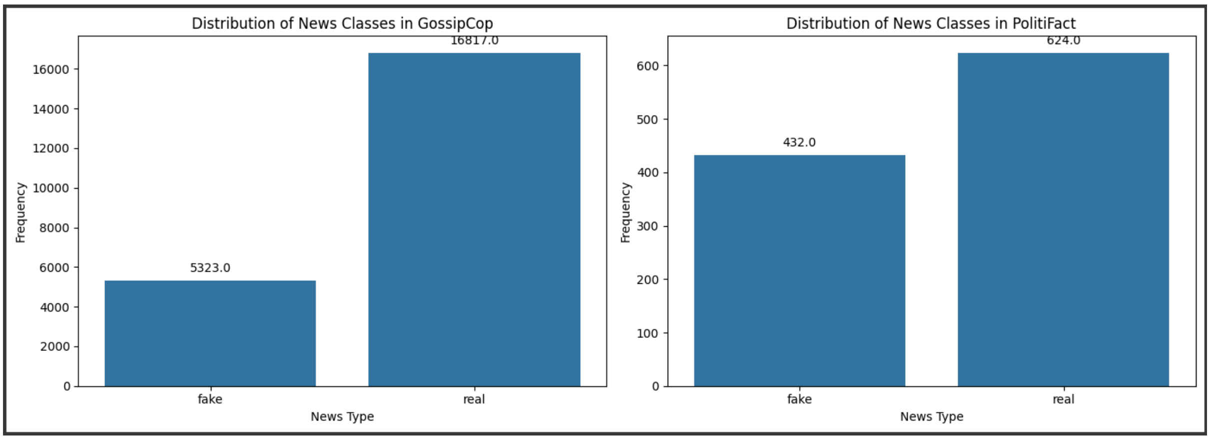


Diagram 4. Text Length Distribution in GossipCop

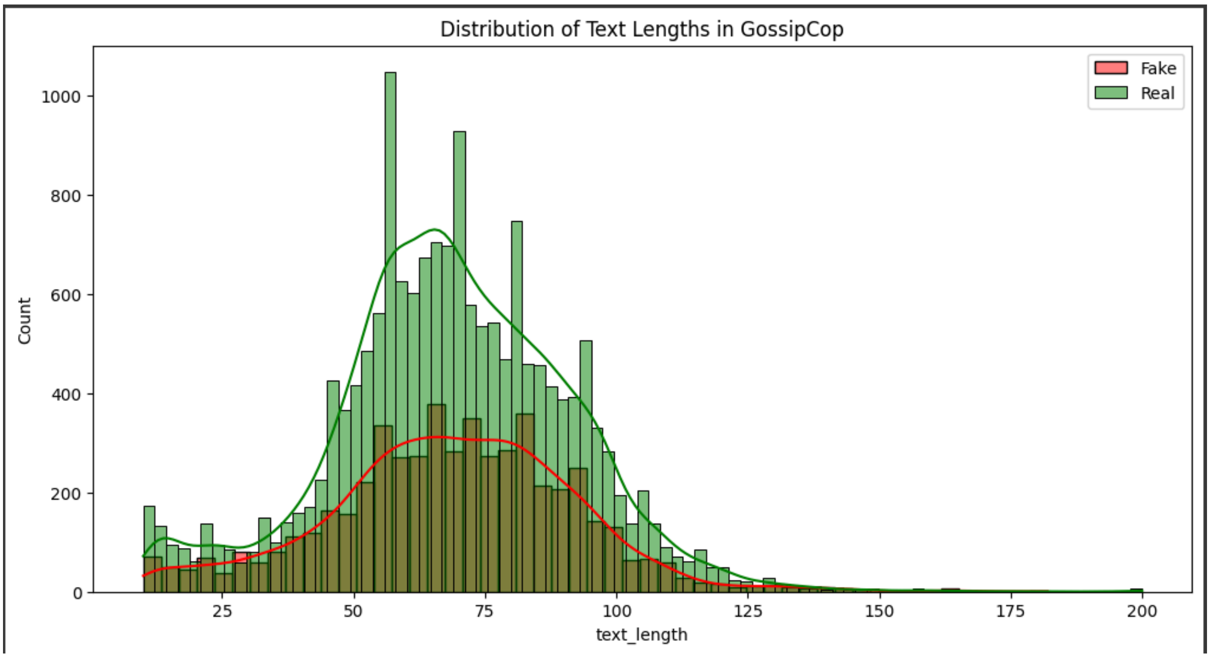


Diagram 5. Text Length Distribution in PolitiFact

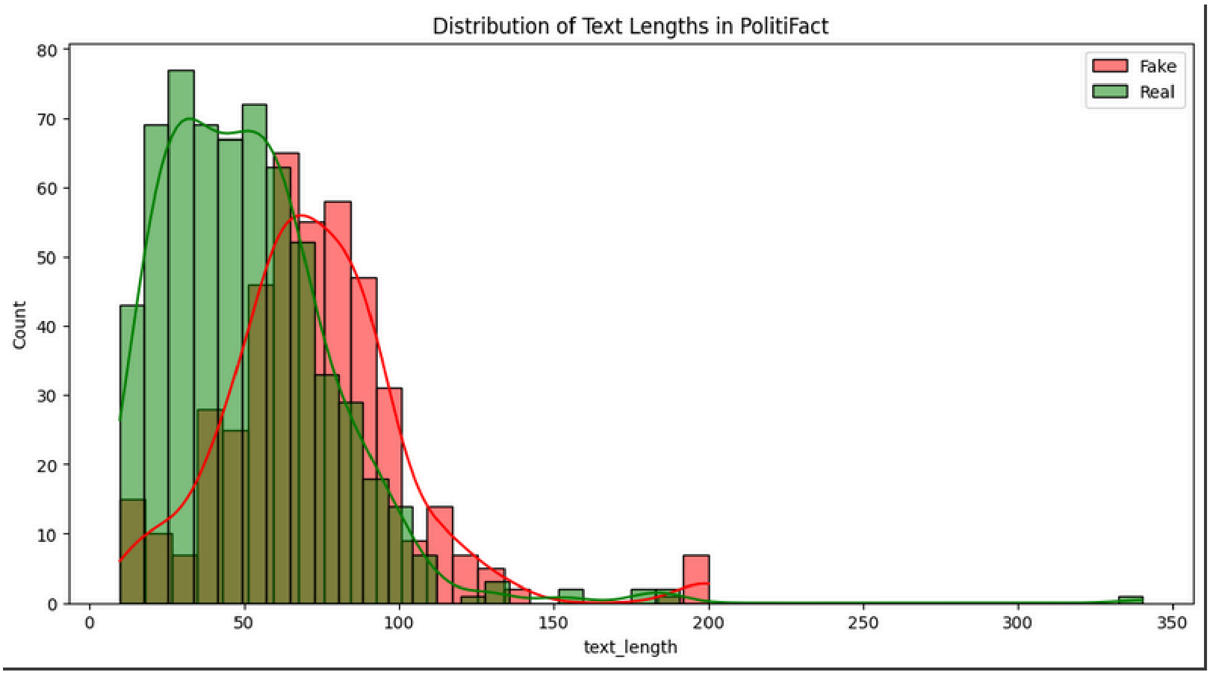


Diagram 6. Word Frequency Analysis

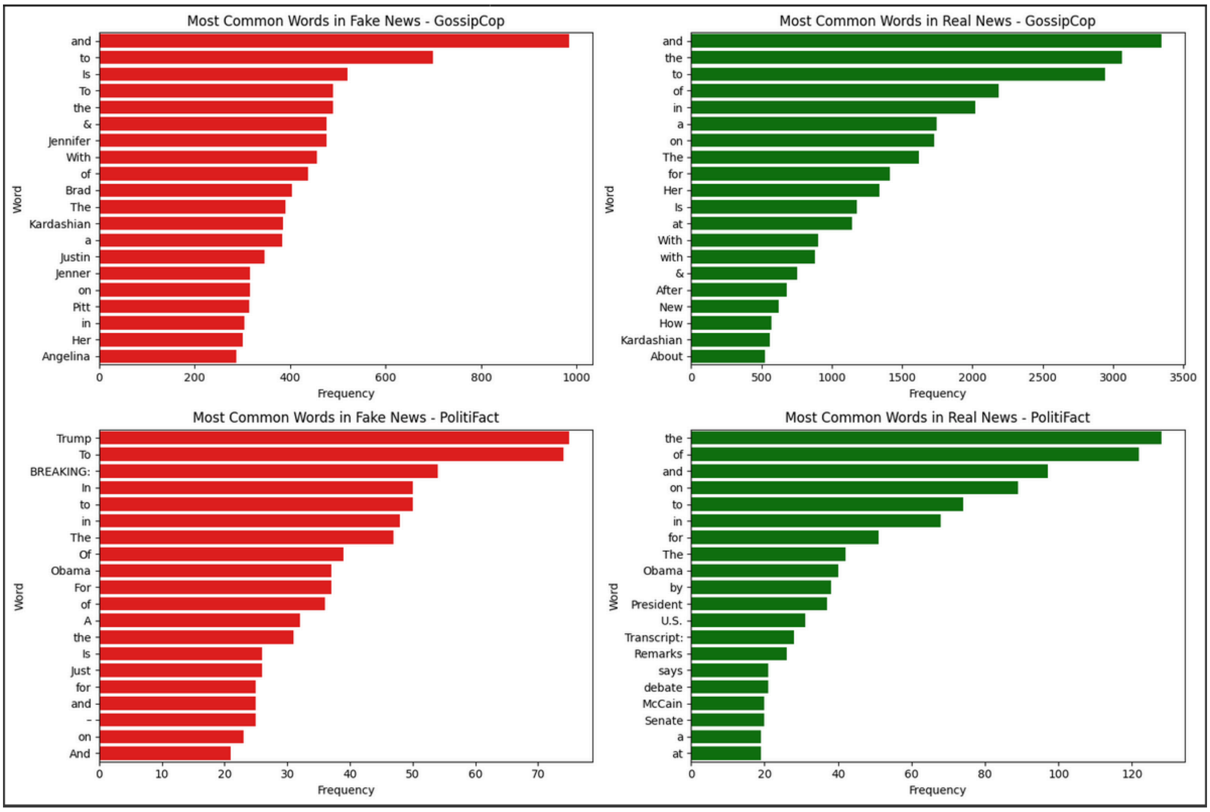


Diagram 7. Most Common Words in Fake GossipCop News

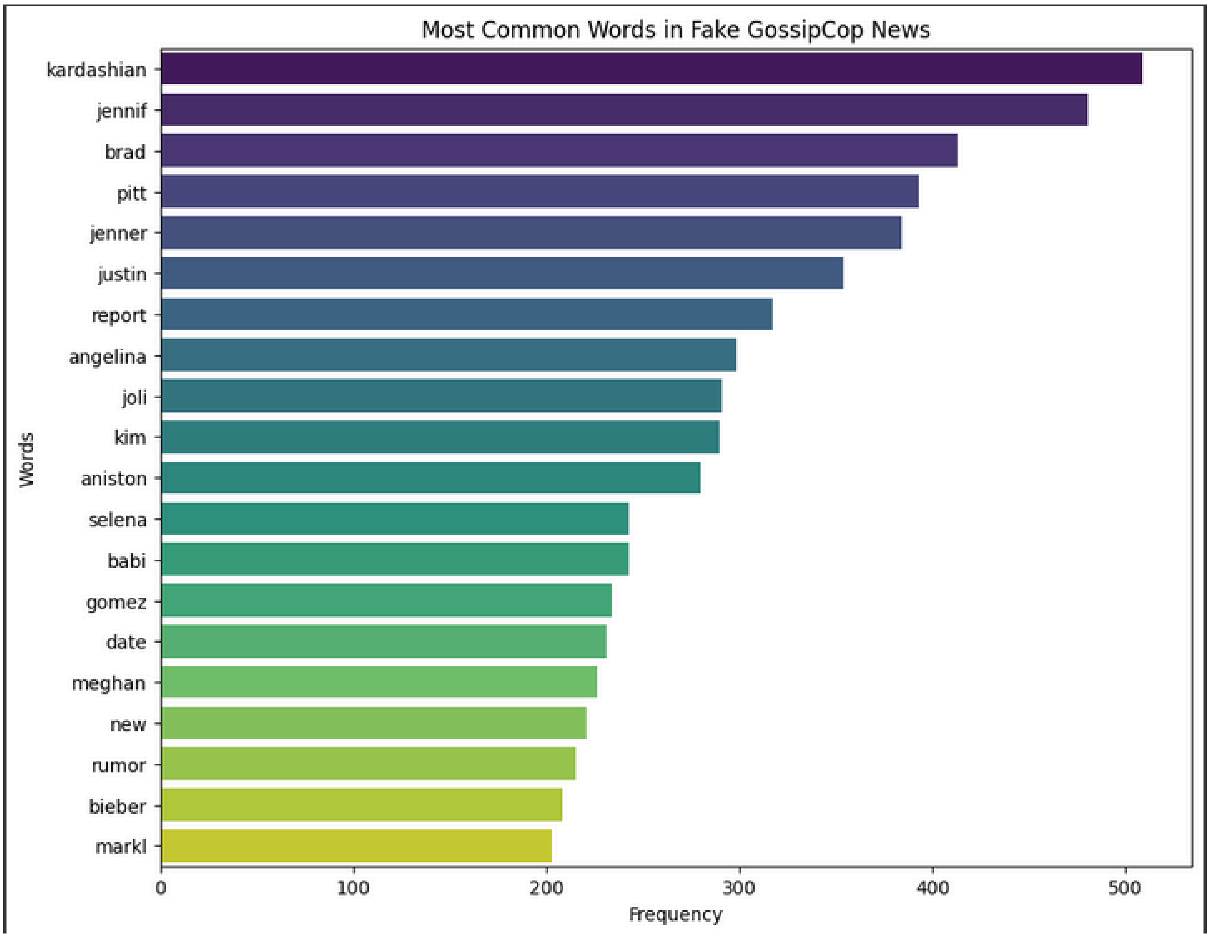


Diagram 8. Most Common Words in Real GossipCop News

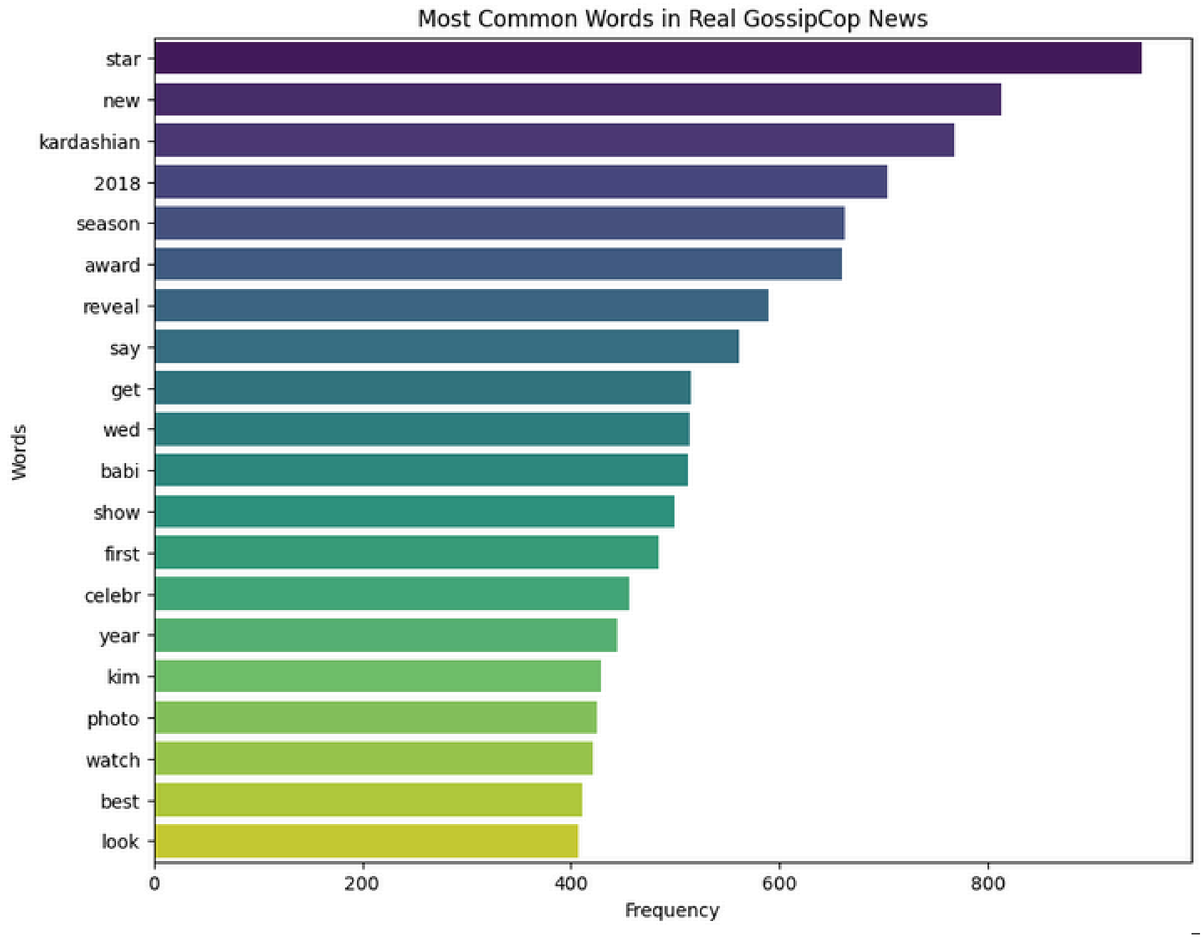


Diagram 9. Most Common Words in Fake PolitiFact News

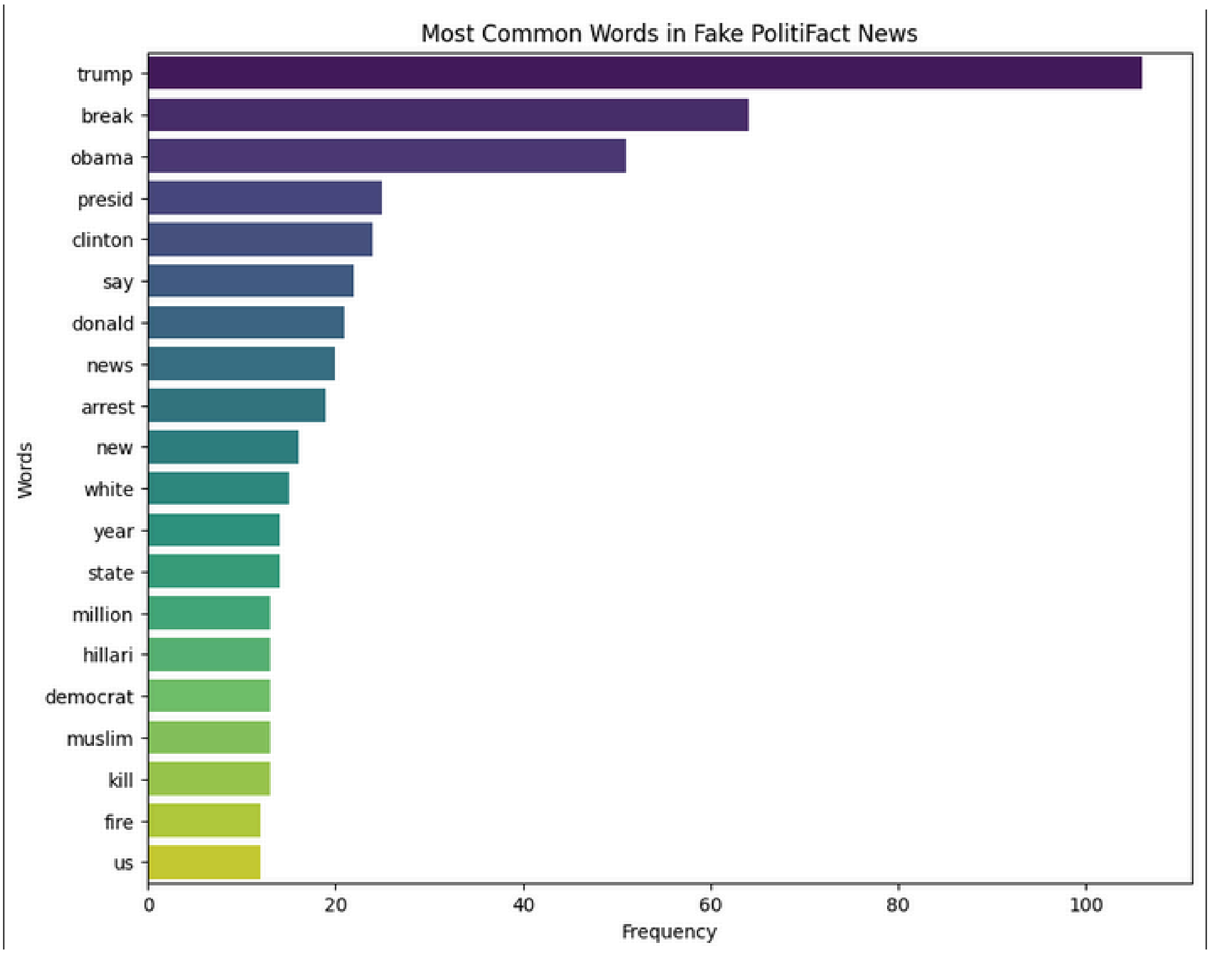


Diagram 10. Most Common Words in Real PolitiFact News

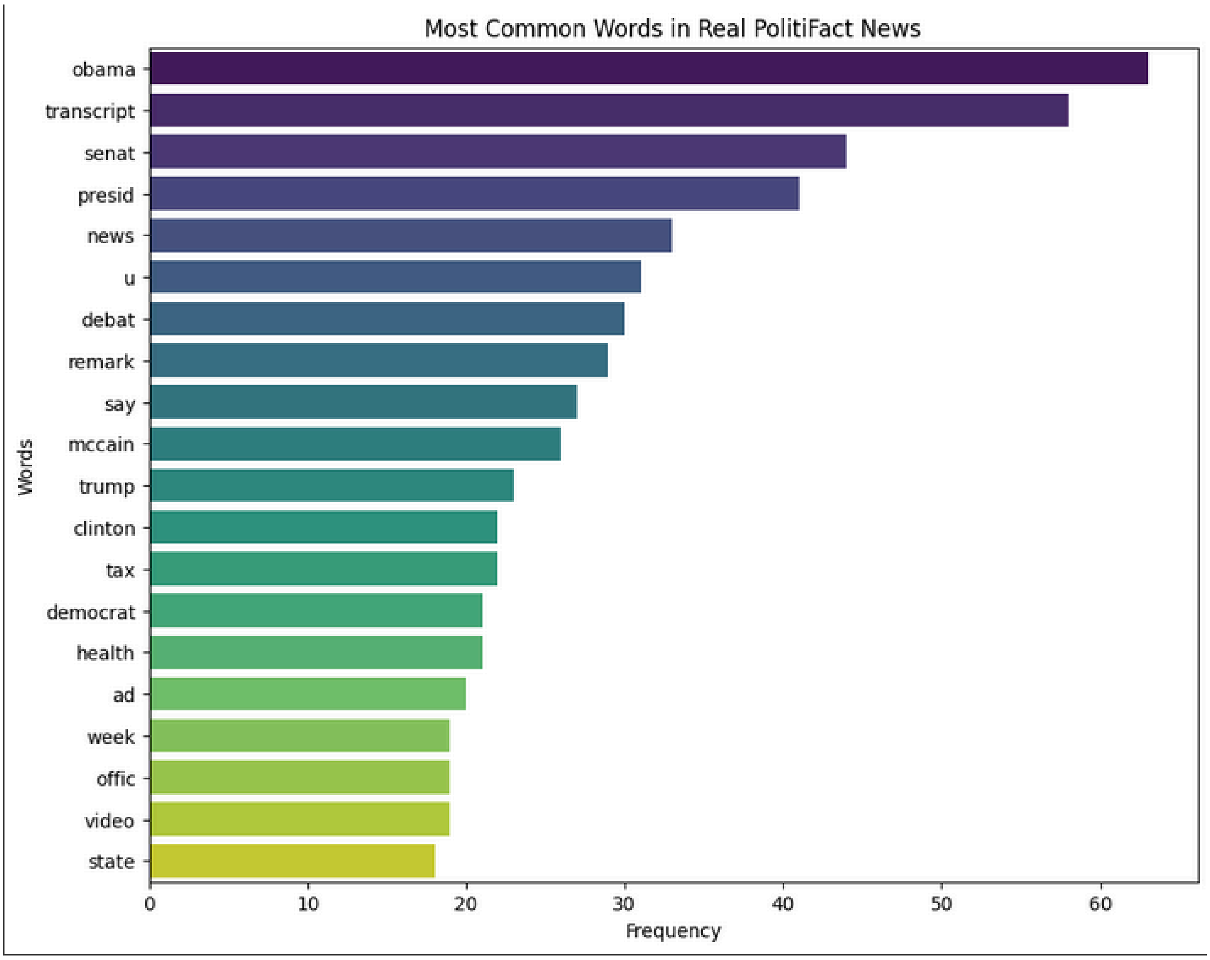


Diagram 11. Best Model SVM GossipCop Confusion Matrix

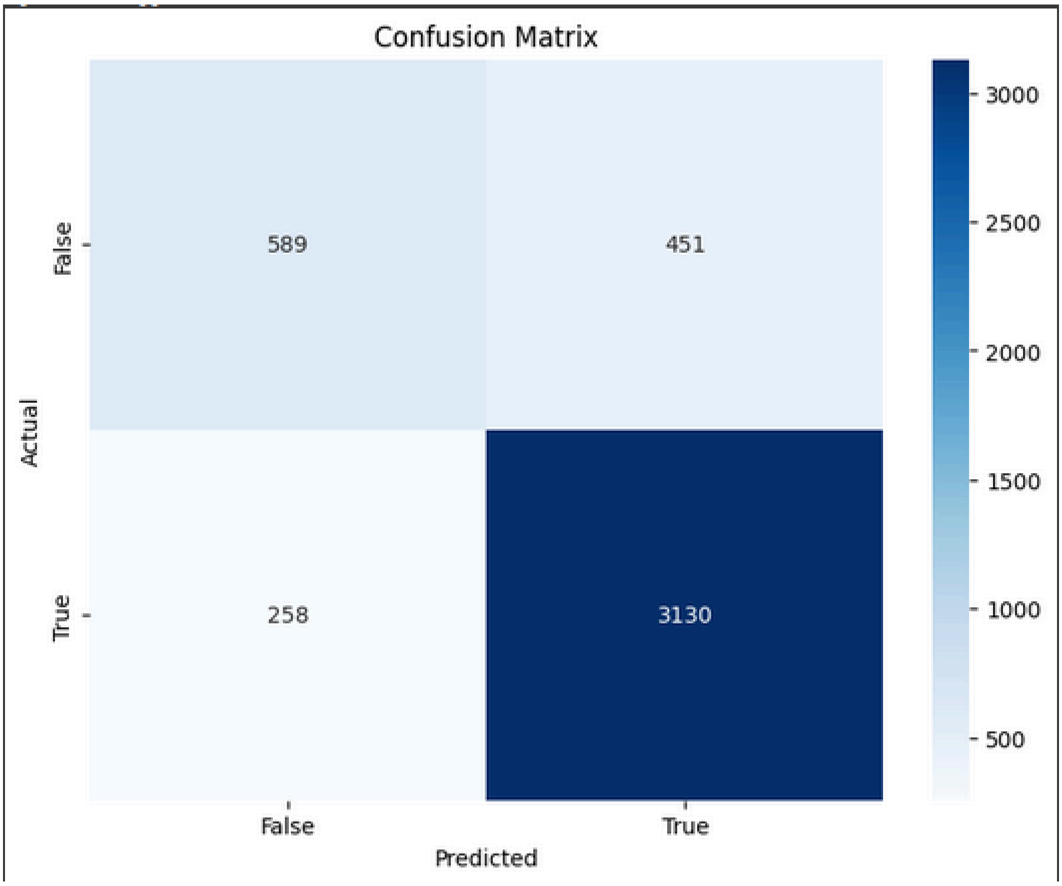


Diagram 12. Best Model SVM GossipCop ROC

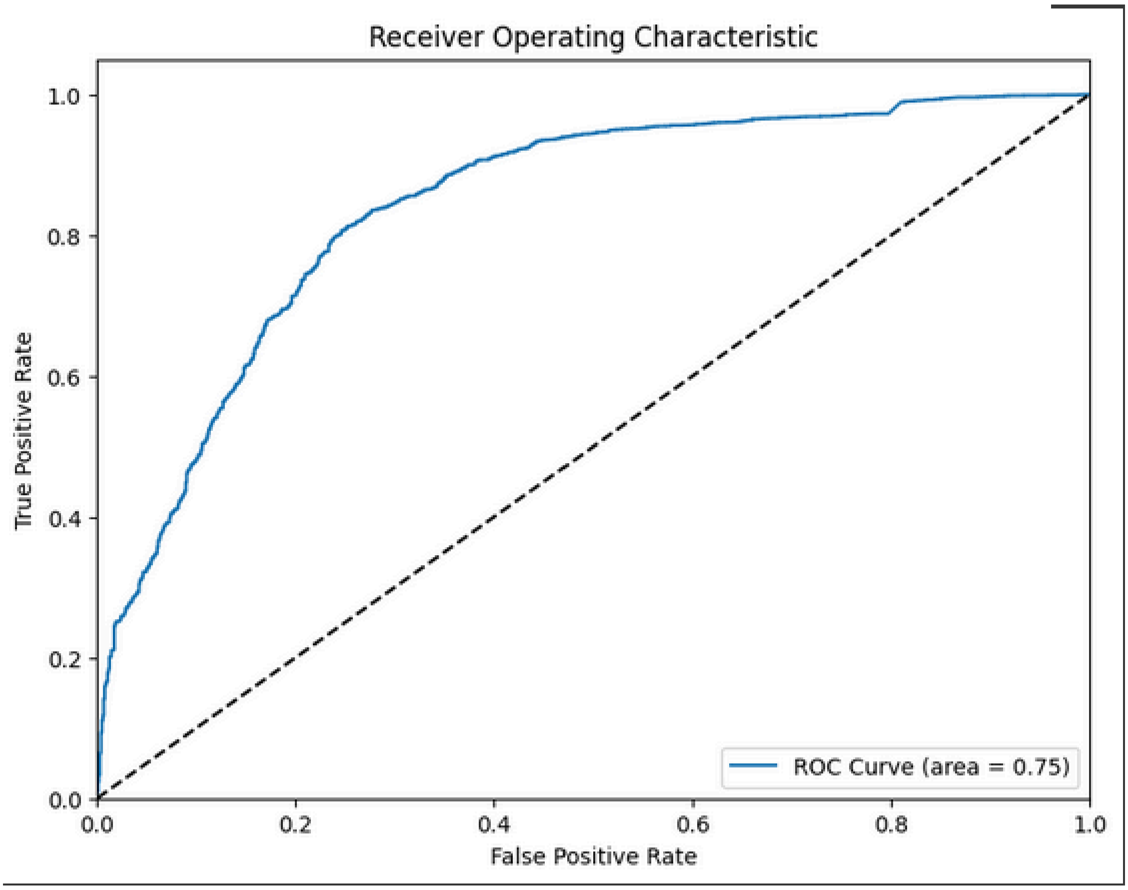
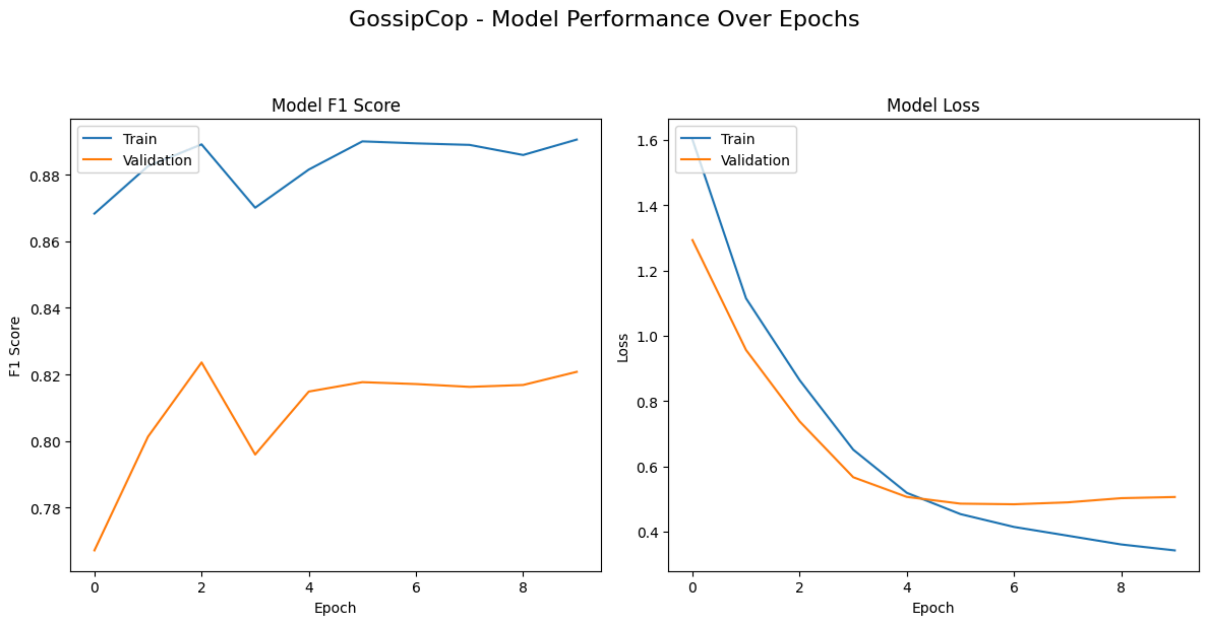
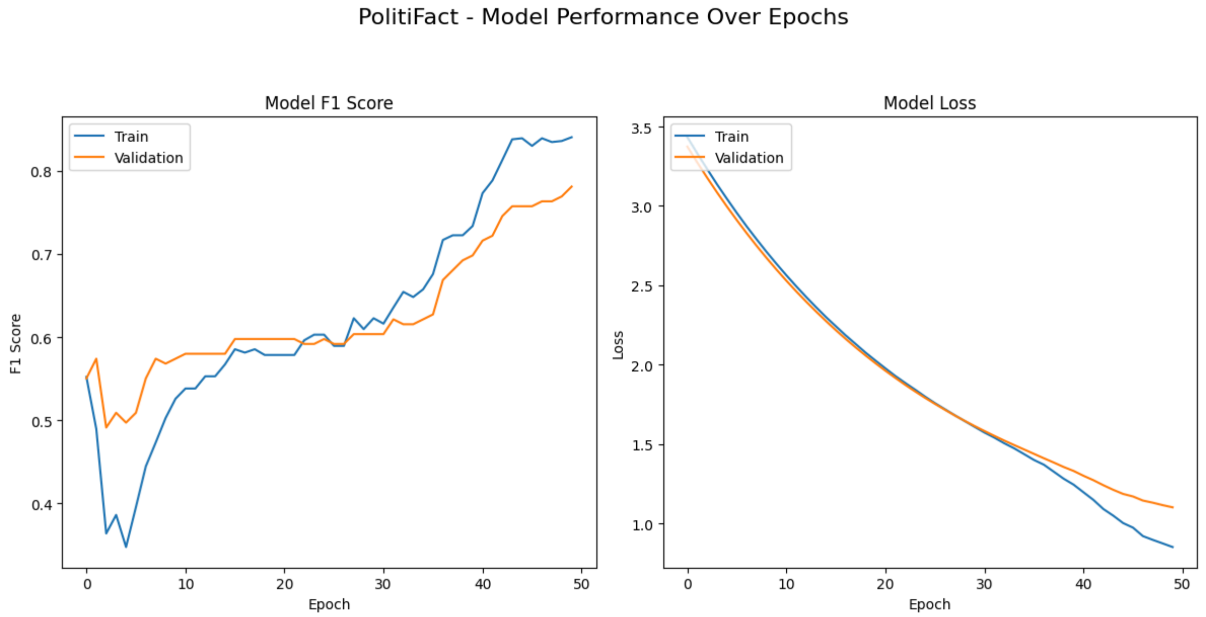
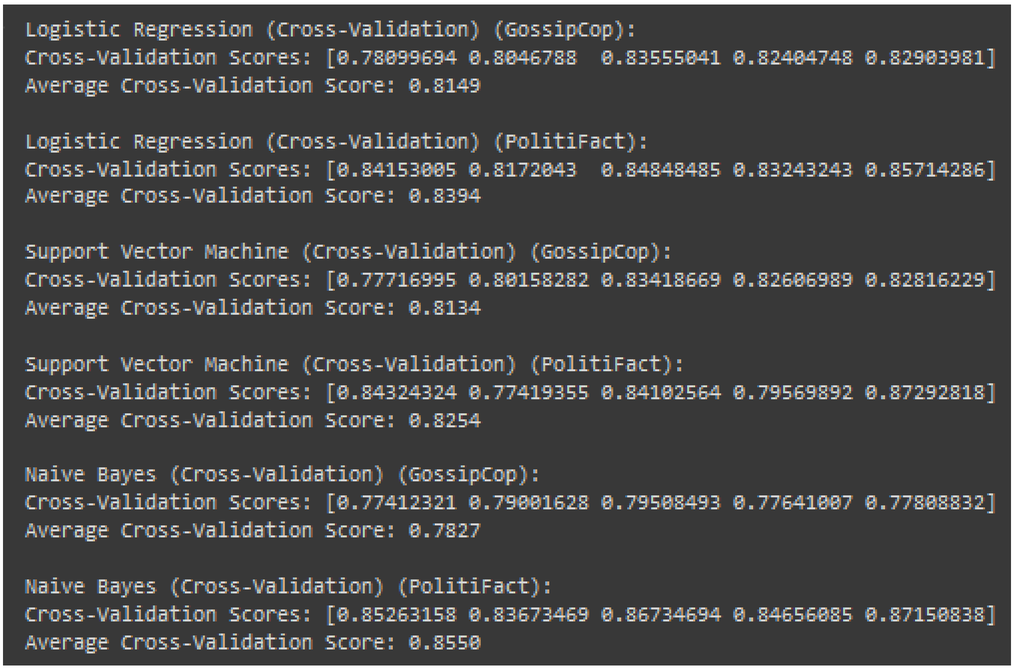


Diagram 13. CNN Model Performance Over Epochs



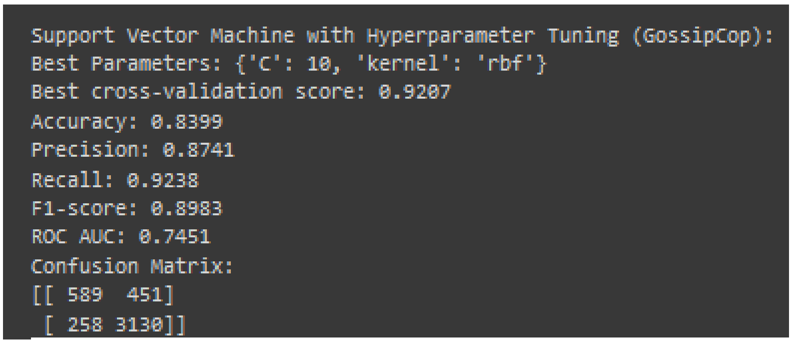


Picture 1. Scores of 5-Fold Cross-Validation



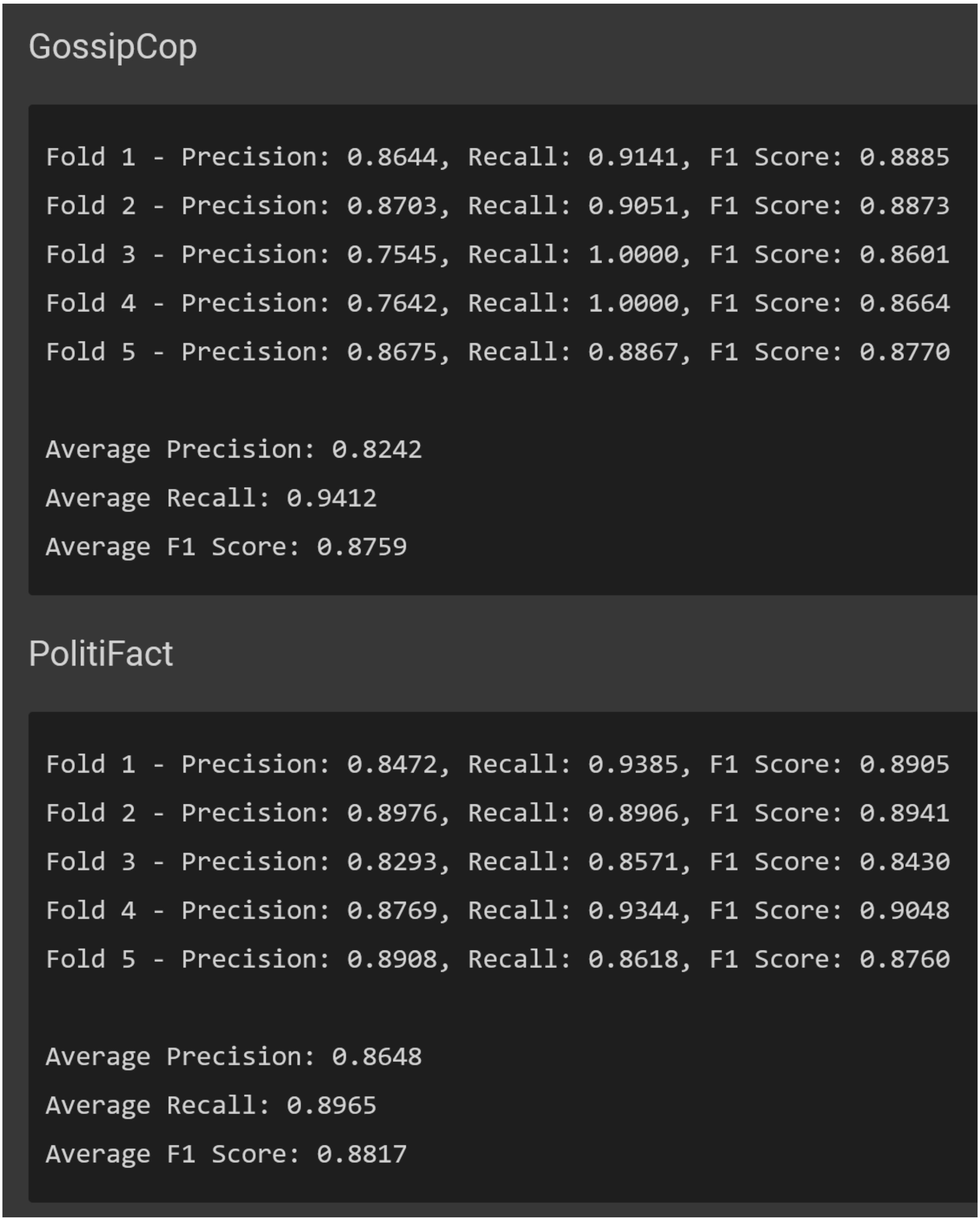
```
Logistic Regression (Cross-Validation) (GossipCop):
Cross-Validation Scores: [0.78099694 0.8046788  0.83555041 0.82404748 0.82903981]
Average Cross-Validation Score: 0.8149

Logistic Regression (Cross-Validation) (PolitiFact):
Cross-Validation Scores: [0.84153005 0.8172043  0.84848485 0.83243243 0.85714286]
Average Cross-Validation Score: 0.8394

Support Vector Machine (Cross-Validation) (GossipCop):
Cross-Validation Scores: [0.77716995 0.80158282 0.83418669 0.82606989 0.82816229]
Average Cross-Validation Score: 0.8134

Support Vector Machine (Cross-Validation) (PolitiFact):
Cross-Validation Scores: [0.84324324 0.77419355 0.84102564 0.79569892 0.87292818]
Average Cross-Validation Score: 0.8254

Naive Bayes (Cross-Validation) (GossipCop):
Cross-Validation Scores: [0.77412321 0.79001628 0.79508493 0.77641007 0.77808832]
Average Cross-Validation Score: 0.7827

Naive Bayes (Cross-Validation) (PolitiFact):
Cross-Validation Scores: [0.85263158 0.83673469 0.86734694 0.84656085 0.87150838]
Average Cross-Validation Score: 0.8550
```

Picture 2: Evaluation statistics of the best-performing model svm_model_gossipcop



```
Support Vector Machine with Hyperparameter Tuning (GossipCop):
Best Parameters: {'C': 10, 'kernel': 'rbf'}
Best cross-validation score: 0.9207
Accuracy: 0.8399
Precision: 0.8741
Recall: 0.9238
F1-score: 0.8983
ROC AUC: 0.7451
Confusion Matrix:
[[ 589  451]
 [ 258 3130]]
```

Picture 3: 5-Fold Cross-Validation results of CNNs



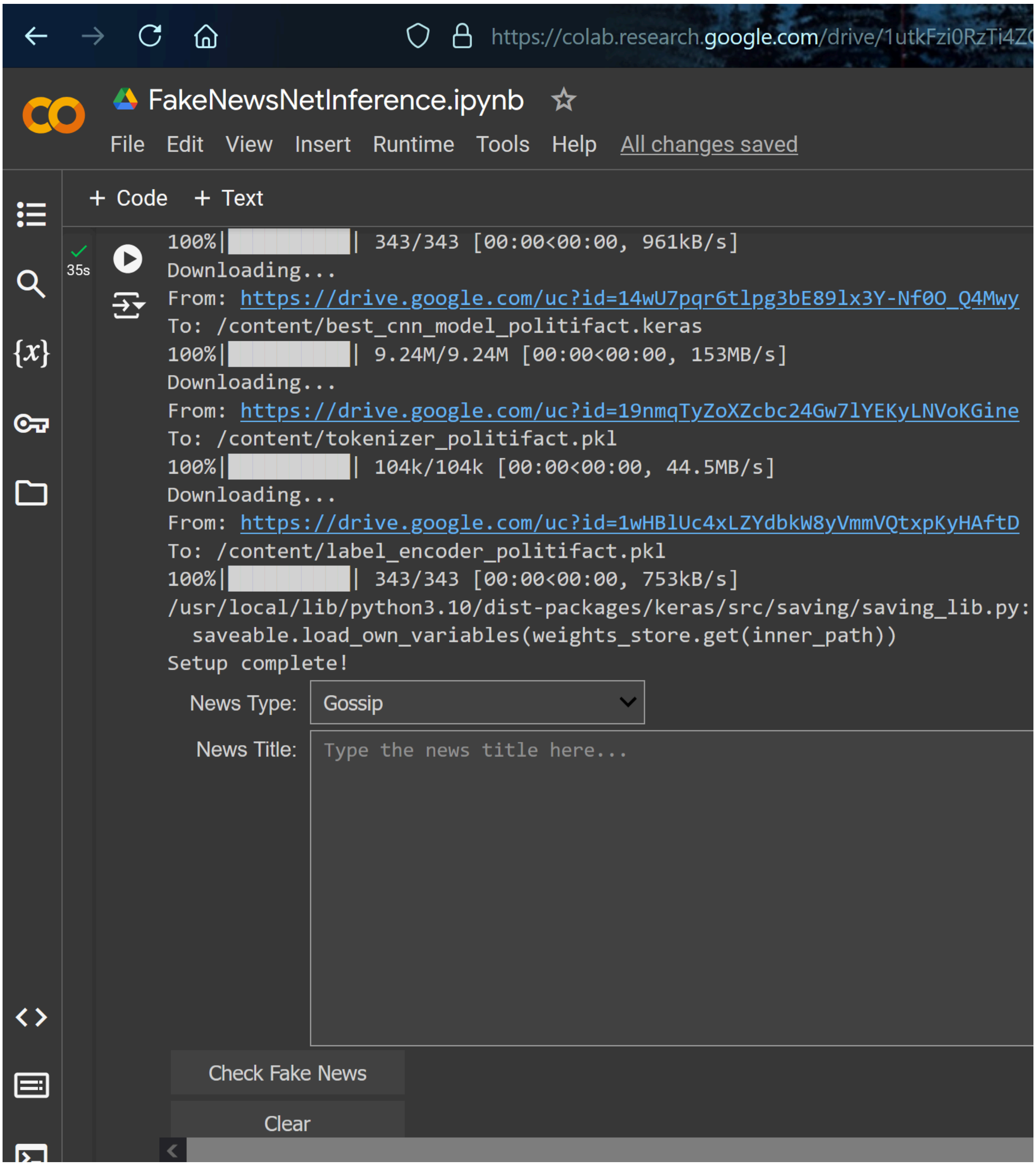
GossipCop

```
Fold 1 - Precision: 0.8644, Recall: 0.9141, F1 Score: 0.8885
Fold 2 - Precision: 0.8703, Recall: 0.9051, F1 Score: 0.8873
Fold 3 - Precision: 0.7545, Recall: 1.0000, F1 Score: 0.8601
Fold 4 - Precision: 0.7642, Recall: 1.0000, F1 Score: 0.8664
Fold 5 - Precision: 0.8675, Recall: 0.8867, F1 Score: 0.8770

Average Precision: 0.8242
Average Recall: 0.9412
Average F1 Score: 0.8759
```

PolitiFact

```
Fold 1 - Precision: 0.8472, Recall: 0.9385, F1 Score: 0.8905
Fold 2 - Precision: 0.8976, Recall: 0.8906, F1 Score: 0.8941
Fold 3 - Precision: 0.8293, Recall: 0.8571, F1 Score: 0.8430
Fold 4 - Precision: 0.8769, Recall: 0.9344, F1 Score: 0.9048
Fold 5 - Precision: 0.8908, Recall: 0.8618, F1 Score: 0.8760

Average Precision: 0.8648
Average Recall: 0.8965
Average F1 Score: 0.8817
```

Picture 4: Inference notebook, user-friendly interface



```
100%|██████████| 343/343 [00:00<00:00, 961kB/s]
Downloading...
From: https://drive.google.com/uc?id=14wU7pqr6tlpg3bE89lx3Y-Nf0O_Q4Mwy
To: /content/best_cnn_model_politifact.keras
100%|██████████| 9.24M/9.24M [00:00<00:00, 153MB/s]
Downloading...
From: https://drive.google.com/uc?id=19nmqTyZoXZcbc24Gw7lYEKyLNVoKGine
To: /content/tokenizer_politifact.pkl
100%|██████████| 104k/104k [00:00<00:00, 44.5MB/s]
Downloading...
From: https://drive.google.com/uc?id=1wHBlUc4xLZYdbkW8yVmmVQtxpKyHAftD
To: /content/label_encoder_politifact.pkl
100%|██████████| 343/343 [00:00<00:00, 753kB/s]
/usr/local/lib/python3.10/dist-packages/keras/src/saving/saving_lib.py:
  saveable.load_own_variables(weights_store.get(inner_path))
Setup complete!
```

News Type: Gossip

News Title: Type the news title here...

The following pages include the PDF versions of the notebooks as appendices for detailed review.

Appendix A: FakeNewsNetPreprocessing
Appendix B: FakeNewsNetClassifier
Appendix C: FakeNewsNetCNN
Appendix D: FakeNewsNetInference

FakeNewsNetPreprocessing

July 14, 2024

1 FakeNewsNetPreprocessing

This notebook preprocesses the FakeNewsNet dataset created by Shu et al. [7]. The dataset is hosted in a GitHub repository[8] and contains political and entertainment news along with social context and spatiotemporal data. The dataset includes the GossipCop and PolitiFact subsets, which will be explored, cleaned, processed, and saved to Google Drive for use in subsequent machine learning tasks in the Classifier notebook.

1.1 Import necessary libraries

[1]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from collections import Counter
import re
from sklearn.utils import shuffle

nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
```

[1]:

```
True
```

1.2 Download the datasets

[2]:

```
# Downloading datasets from the GitHub repository
!wget "https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/master/dataset/
 ↪gossipcop_fake.csv" -O gossipcop_fake.csv
!wget "https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/master/dataset/
 ↪gossipcop_real.csv" -O gossipcop_real.csv
!wget "https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/master/dataset/
 ↪politifact_fake.csv" -O politifact_fake.csv
!wget "https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/master/dataset/
 ↪politifact_real.csv" -O politifact_real.csv
```

```
--2024-07-14 07:18:28--  https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/m
aster/dataset/gossipcop_fake.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.108.133, 185.199.109.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 12538704 (12M) [text/plain]
Saving to: ‘gossipcop_fake.csv’

gossipcop_fake.csv  100%[===================>]  11.96M  79.1MB/s    in 0.2s

2024-07-14 07:18:28 (79.1 MB/s) - ‘gossipcop_fake.csv’ saved [12538704/12538704]

--2024-07-14 07:18:28--  https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/m
aster/dataset/gossipcop_real.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.111.133, 185.199.109.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.111.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 19978126 (19M) [text/plain]
Saving to: ‘gossipcop_real.csv’

gossipcop_real.csv  100%[===================>]  19.05M  96.8MB/s    in 0.2s

2024-07-14 07:18:28 (96.8 MB/s) - ‘gossipcop_real.csv’ saved [19978126/19978126]

--2024-07-14 07:18:28--  https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/m
aster/dataset/politifact_fake.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.109.133, 185.199.110.133, 185.199.108.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.109.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 3286418 (3.1M) [text/plain]
Saving to: ‘politifact_fake.csv’
```

```
politifact_fake.csv 100%[===================>]   3.13M  --.-KB/s    in 0.08s

2024-07-14 07:18:29 (38.7 MB/s) - ‘politifact_fake.csv’ saved [3286418/3286418]

--2024-07-14 07:18:29--  https://raw.githubusercontent.com/KaiDMML/FakeNewsNet/m
aster/dataset/politifact_real.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.109.133, 185.199.111.133, 185.199.108.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.109.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 8278658 (7.9M) [text/plain]
Saving to: ‘politifact_real.csv’

politifact_real.csv 100%[===================>]   7.89M  52.3MB/s    in 0.2s

2024-07-14 07:18:29 (52.3 MB/s) - ‘politifact_real.csv’ saved [8278658/8278658]
```

1.3 Load the Data

[3]:
```
# Load the datasets
gossipcop_fake = pd.read_csv('gossipcop_fake.csv')
gossipcop_real = pd.read_csv('gossipcop_real.csv')
politifact_fake = pd.read_csv('politifact_fake.csv')
politifact_real = pd.read_csv('politifact_real.csv')

# Quick check of the data loaded
print(gossipcop_fake.head())
print(gossipcop_real.head())
print(politifact_fake.head())
print(politifact_real.head())
```

```
                     id                                           news_url  \
0  gossipcop-2493749932  www.dailymail.co.uk/tvshowbiz/article-5874213/…
1  gossipcop-4580247171  hollywoodlife.com/2018/05/05/paris-jackson-car…
2   gossipcop-941805037  variety.com/2017/biz/news/tax-march-donald-tru…
3  gossipcop-2547891536  www.dailymail.co.uk/femail/article-3499192/Do-…
4  gossipcop-5476631226  variety.com/2018/film/news/list-2018-oscar-nom…

                                               title  \
0  Did Miley Cyrus and Liam Hemsworth secretly ge…
1  Paris Jackson & Cara Delevingne Enjoy Night Ou…
2  Celebrities Join Tax March in Protest of Donal…
3  Cindy Crawford's daughter Kaia Gerber wears a …
4      Full List of 2018 Oscar Nominations - Variety
```

```
                                           tweet_ids
0  284329075902926848\t284332744559968256\t284335…
1  992895508267130880\t992897935418503169\t992899…
2  853359353532829696\t853359576543920128\t853359…
3  988821905196158981\t988824206556172288\t988825…
4  955792793632432131\t955795063925301249\t955798…
                 id                                           news_url  \
0  gossipcop-882573  https://www.brides.com/story/teen-mom-jenelle-…
1  gossipcop-875924  https://www.dailymail.co.uk/tvshowbiz/article-…
2  gossipcop-894416        https://en.wikipedia.org/wiki/Quinn_Perkins
3  gossipcop-857248  https://www.refinery29.com/en-us/2018/03/19192…
4  gossipcop-884684  https://www.cnn.com/2017/10/04/entertainment/c…

                                               title  \
0  Teen Mom Star Jenelle Evans' Wedding Dress Is …
1  Kylie Jenner refusing to discuss Tyga on Life …
2                                      Quinn Perkins
3  I Tried Kim Kardashian's Butt Workout & Am For…
4  Celine Dion donates concert proceeds to Vegas …

                                           tweet_ids
0  912371411146149888\t912371528343408641\t912372…
1  901989917546426369\t901989992074969089\t901990…
2  931263637246881792\t931265332022579201\t931265…
3  868114761723936769\t868122567910936576\t868128…
4  915528047004209152\t915529285171122176\t915530…
                id                                           news_url  \
0  politifact15014          speedtalk.com/forum/viewtopic.php?t=51650
1  politifact15156  politics2020.info/index.php/2018/03/13/court-o…
2  politifact14745  www.nscdscamps.org/blog/category/parenting/467…
3  politifact14355  https://howafrica.com/oscar-pistorius-attempts…
4  politifact15371  http://washingtonsources.org/trump-votes-for-d…

                                               title  \
0  BREAKING: First NFL Team Declares Bankruptcy O…
1  Court Orders Obama To Pay $400 Million In Rest…
2  UPDATE: Second Roy Moore Accuser Works For Mic…
3         Oscar Pistorius Attempts To Commit Suicide
4        Trump Votes For Death Penalty For Being Gay

                                           tweet_ids
0  937349434668498944\t937379378006282240\t937380…
1  972666281441878016\t972678396575559680\t972827…
2  929405740732870656\t929439450400264192\t929439…
3  886941526458347521\t887011300278194176\t887023…
4  915205698212040704\t915242076681506816\t915249…
                id                                           news_url  \
0  politifact14984                          http://www.nfib-sbet.org/
```

```
1  politifact12944  http://www.cq.com/doc/newsmakertranscripts-494…
2    politifact333  https://web.archive.org/web/20080204072132/htt…
3   politifact4358  https://web.archive.org/web/20110811143753/htt…
4    politifact779  https://web.archive.org/web/20070820164107/htt…

                                               title  \
0        National Federation of Independent Business
1                            comments in Fayetteville NC
2  Romney makes pitch, hoping to close deal : Ele…
3  Democratic Leaders Say House Democrats Are Uni…
4    Budget of the United States Government, FY 2008

                                           tweet_ids
0  967132259869487105\t967164368768196609\t967215…
1  942953459\t8980098198\t16253717352\t1668513250…
2                                                NaN
3                                                NaN
4  89804710374154240\t91270460595109888\t96039619…
```

1.4 Data Inspection

[4]:
```python
# Display basic information about each dataset
gossipcop_fake.info()
gossipcop_real.info()
politifact_fake.info()
politifact_real.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5323 entries, 0 to 5322
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         5323 non-null   object
 1   news_url   5067 non-null   object
 2   title      5323 non-null   object
 3   tweet_ids  5135 non-null   object
dtypes: object(4)
memory usage: 166.5+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16817 entries, 0 to 16816
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         16817 non-null  object
 1   news_url   16804 non-null  object
 2   title      16817 non-null  object
 3   tweet_ids  15759 non-null  object
dtypes: object(4)
```

```
memory usage: 525.7+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 432 entries, 0 to 431
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         432 non-null    object
 1   news_url   428 non-null    object
 2   title      432 non-null    object
 3   tweet_ids  392 non-null    object
dtypes: object(4)
memory usage: 13.6+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 624 entries, 0 to 623
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         624 non-null    object
 1   news_url   567 non-null    object
 2   title      624 non-null    object
 3   tweet_ids  409 non-null    object
dtypes: object(4)
memory usage: 19.6+ KB
```

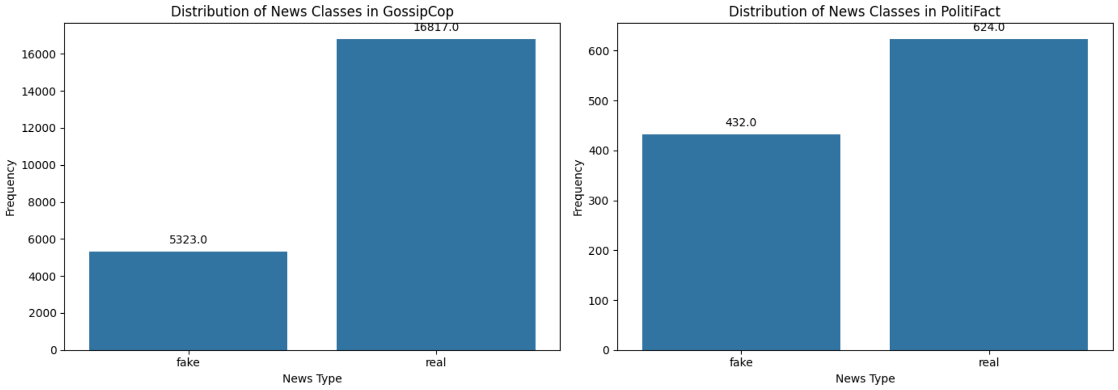
1.5 Initial Exploratory Data Analysis (EDA)

1.5.1 Visualization of Class Distribution

[5]:

```python
# Label the data
gossipcop_fake['label'] = 'fake'
gossipcop_real['label'] = 'real'
politifact_fake['label'] = 'fake'
politifact_real['label'] = 'real'

# Add dataset identifier
gossipcop_fake['dataset'] = 'GossipCop'
gossipcop_real['dataset'] = 'GossipCop'
politifact_fake['dataset'] = 'PolitiFact'
politifact_real['dataset'] = 'PolitiFact'

# Combine the datasets
combined = pd.concat([
    gossipcop_fake[['label', 'dataset']],
    gossipcop_real[['label', 'dataset']],
    politifact_fake[['label', 'dataset']],
    politifact_real[['label', 'dataset']]
])
```

```
# Plotting
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Plot for GossipCop
gossipcop_plot = sns.countplot(x='label', data=combined[combined['dataset'] ==␣
  ↪'GossipCop'], ax=axes[0])
axes[0].set_title('Distribution of News Classes in GossipCop')
axes[0].set_xlabel('News Type')
axes[0].set_ylabel('Frequency')
# Annotate bars with count
for p in gossipcop_plot.patches:
    gossipcop_plot.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2.
  ↪, p.get_height()), ha = 'center', va = 'center', xytext = (0, 10),␣
  ↪textcoords = 'offset points')

# Plot for PolitiFact
politifact_plot = sns.countplot(x='label', data=combined[combined['dataset'] ==␣
  ↪'PolitiFact'], ax=axes[1])
axes[1].set_title('Distribution of News Classes in PolitiFact')
axes[1].set_xlabel('News Type')
axes[1].set_ylabel('Frequency')
# Annotate bars with count
for p in politifact_plot.patches:
    politifact_plot.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() /␣
  ↪2., p.get_height()), ha = 'center', va = 'center', xytext = (0, 10),␣
  ↪textcoords = 'offset points')

plt.tight_layout()
plt.show()
```

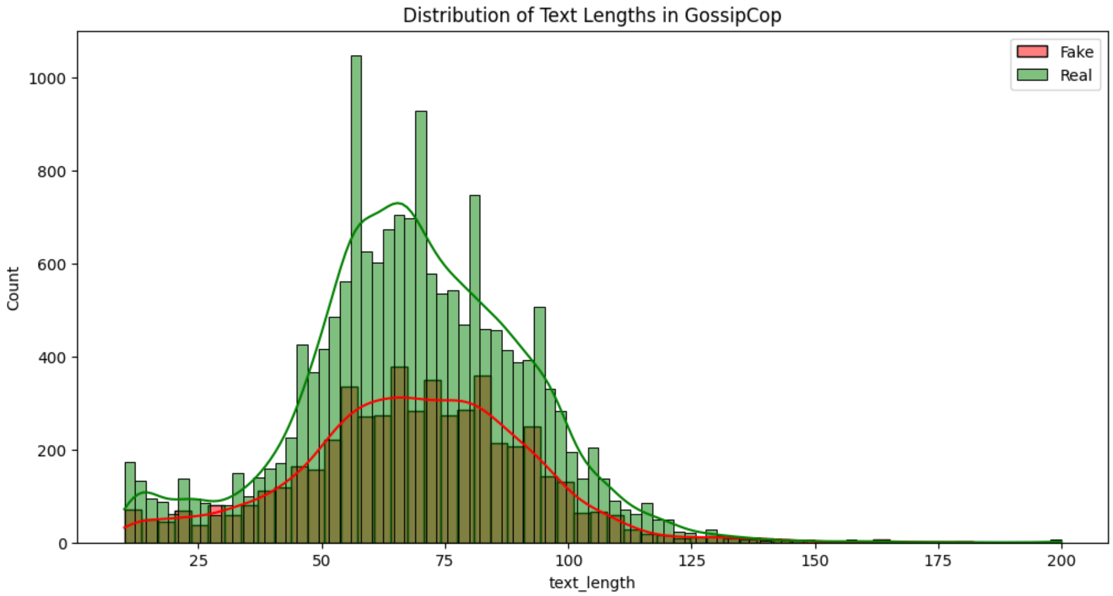


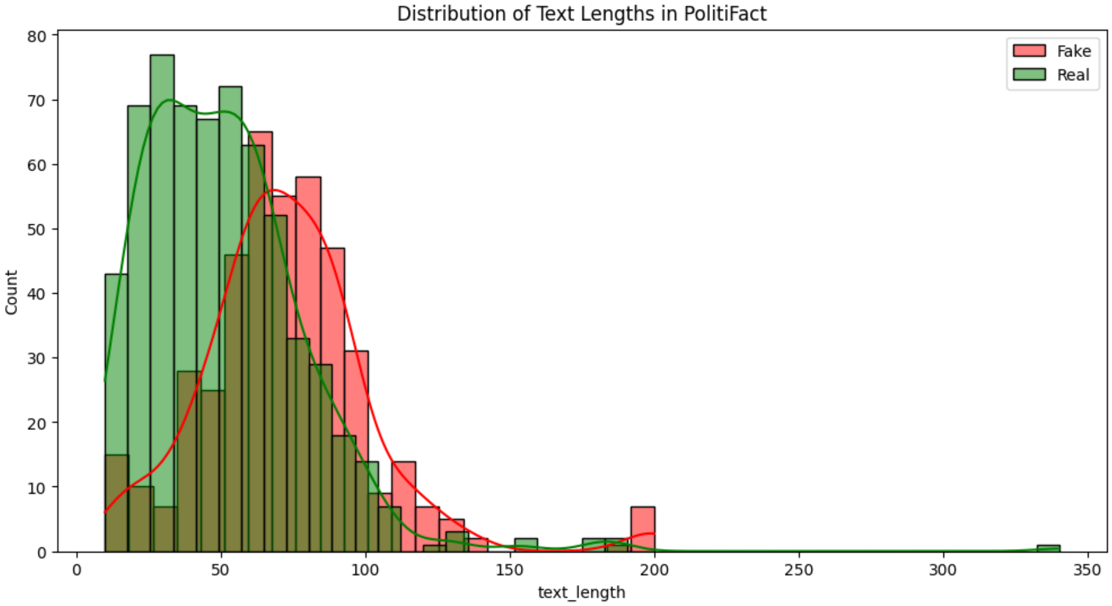
The datasets have class imbalance. There are approximately 3 times more records of real news than fake news in the GossipCop dataset. In the PolitiFact dataset, the number of fake news articles is about 30% less than that of real news articles. This will require a strategy to balance the dataset.

1.5.2 Text Length Distribution

I will check text length distribution in title column of both datasets and compare the real and fake news.

```
[6]: # Add text length as a new feature
gossipcop_fake['text_length'] = gossipcop_fake['title'].apply(len)
gossipcop_real['text_length'] = gossipcop_real['title'].apply(len)
politifact_fake['text_length'] = politifact_fake['title'].apply(len)
politifact_real['text_length'] = politifact_real['title'].apply(len)

# Visualize the distribution of text length
plt.figure(figsize=(12, 6))
sns.histplot(gossipcop_fake['text_length'], color='red', label='Fake', kde=True)
sns.histplot(gossipcop_real['text_length'], color='green', label='Real',␣
 ↪kde=True)
plt.title('Distribution of Text Lengths in GossipCop')
plt.legend()
plt.show()

plt.figure(figsize=(12, 6))
sns.histplot(politifact_fake['text_length'], color='red', label='Fake',␣
 ↪kde=True)
sns.histplot(politifact_real['text_length'], color='green', label='Real',␣
 ↪kde=True)
plt.title('Distribution of Text Lengths in PolitiFact')
plt.legend()
plt.show()
```





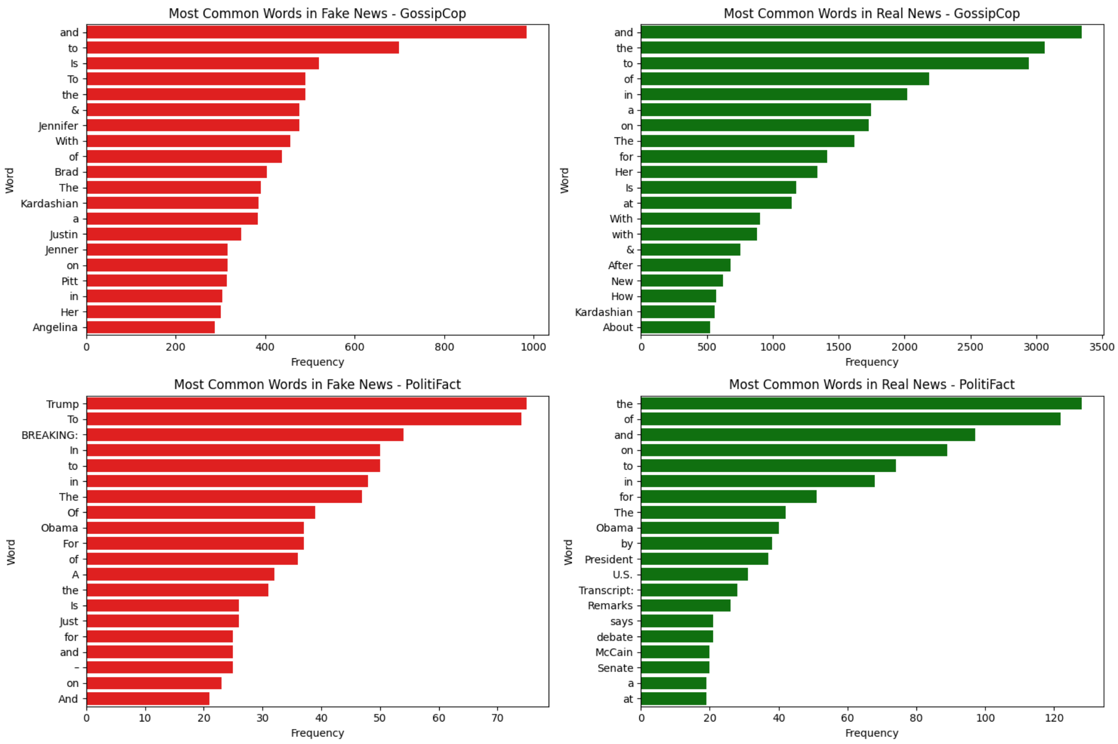
The text length distributions indicate that real news generally has shorter titles compared to fake news in both GossipCop and PolitiFact datasets.

1.5.3 Word Frequency Analysis

[7]:

```
# Function to get the most common words
def get_most_common_words(text_list, num=20):
    all_words = ' '.join(text_list).split()
    common_words = Counter(all_words).most_common(num)
    return pd.DataFrame(common_words, columns=['Word', 'Frequency'])

# Apply the function to each dataset
fake_words_gossipcop = get_most_common_words(gossipcop_fake['title'])
real_words_gossipcop = get_most_common_words(gossipcop_real['title'])
fake_words_politifact = get_most_common_words(politifact_fake['title'])
real_words_politifact = get_most_common_words(politifact_real['title'])

# Plotting the results
fig, axes = plt.subplots(2, 2, figsize=(15, 10))

sns.barplot(x='Frequency', y='Word', data=fake_words_gossipcop, ax=axes[0, 0],␣
 ↪color='red')
axes[0, 0].set_title('Most Common Words in Fake News - GossipCop')

sns.barplot(x='Frequency', y='Word', data=real_words_gossipcop, ax=axes[0, 1],␣
 ↪color='green')
axes[0, 1].set_title('Most Common Words in Real News - GossipCop')
```

```
sns.barplot(x='Frequency', y='Word', data=fake_words_politifact, ax=axes[1, 0],␣
  ↪color='red')
axes[1, 0].set_title('Most Common Words in Fake News - PolitiFact')

sns.barplot(x='Frequency', y='Word', data=real_words_politifact, ax=axes[1, 1],␣
  ↪color='green')
axes[1, 1].set_title('Most Common Words in Real News - PolitiFact')

plt.tight_layout()
plt.show()
```



The most common words in fake news GossipCop are: and, to, Is. The most common words in real news GossipCop are: and, the, to. The most common words in fake news PolitiFact are: Trump, To, BREAKING:. The most common words in real news PolitiFact are: the, of, and. The results contain stopwords and punctuation and text in title column will need preprocessing.

1.6 Data Cleaning

I will clean the data by removing unnecessary columns, which will prepare the datasets to include only relevant features (text column) for model training. Most machine learning algorithms require input data, including labels, to be numeric, so I will convert labels (label column) to numeric values. I will check for missing values in all columns of each dataset which will ensure the data's integrity. Cleaning step will ensure the compatibility with the machine learning algorithms that will be used in classifier notebook.

[8]:

```
# Drop unneeded columns
columns_to_drop = ['id', 'news_url', 'tweet_ids', 'text_length', 'dataset']
gossipcop_fake.drop(columns=columns_to_drop, inplace=True)
gossipcop_real.drop(columns=columns_to_drop, inplace=True)
politifact_fake.drop(columns=columns_to_drop, inplace=True)
politifact_real.drop(columns=columns_to_drop, inplace=True)

# Convert labels to numeric values
label_mapping = {'fake': 0, 'real': 1}
gossipcop_fake['label'] = gossipcop_fake['label'].map(label_mapping)
gossipcop_real['label'] = gossipcop_real['label'].map(label_mapping)
politifact_fake['label'] = politifact_fake['label'].map(label_mapping)
politifact_real['label'] = politifact_real['label'].map(label_mapping)

# Check for missing values in all columns of each dataset
missing_values_gossipcop_fake = gossipcop_fake.isnull().sum()
missing_values_gossipcop_real = gossipcop_real.isnull().sum()
missing_values_politifact_fake = politifact_fake.isnull().sum()
missing_values_politifact_real = politifact_real.isnull().sum()

print("Missing values in GossipCop Fake:\n", missing_values_gossipcop_fake)
print("\nMissing values in GossipCop Real:\n", missing_values_gossipcop_real)
print("\nMissing values in PolitiFact Fake:\n", missing_values_politifact_fake)
print("\nMissing values in PolitiFact Real:\n", missing_values_politifact_real)

# check the datasets
print(gossipcop_fake.head())
print(gossipcop_real.head())
print(politifact_fake.head())
print(politifact_real.head())
```

```
Missing values in GossipCop Fake:
 title    0
label    0
dtype: int64

Missing values in GossipCop Real:
 title    0
label    0
dtype: int64

Missing values in PolitiFact Fake:
 title    0
label    0
dtype: int64
```

```
Missing values in PolitiFact Real:
 title    0
label    0
dtype: int64
                                               title  label
0  Did Miley Cyrus and Liam Hemsworth secretly ge…      0
1  Paris Jackson & Cara Delevingne Enjoy Night Ou…      0
2  Celebrities Join Tax March in Protest of Donal…      0
3  Cindy Crawford's daughter Kaia Gerber wears a …      0
4      Full List of 2018 Oscar Nominations - Variety      0
                                               title  label
0  Teen Mom Star Jenelle Evans' Wedding Dress Is …      1
1  Kylie Jenner refusing to discuss Tyga on Life …      1
2                                      Quinn Perkins      1
3  I Tried Kim Kardashian's Butt Workout & Am For…      1
4  Celine Dion donates concert proceeds to Vegas …      1
                                               title  label
0  BREAKING: First NFL Team Declares Bankruptcy O…      0
1  Court Orders Obama To Pay $400 Million In Rest…      0
2  UPDATE: Second Roy Moore Accuser Works For Mic…      0
3         Oscar Pistorius Attempts To Commit Suicide      0
4        Trump Votes For Death Penalty For Being Gay      0
                                               title  label
0        National Federation of Independent Business      1
1                        comments in Fayetteville NC      1
2  Romney makes pitch, hoping to close deal : Ele…      1
3  Democratic Leaders Say House Democrats Are Uni…      1
4    Budget of the United States Government, FY 2008      1
```

1.7 Feature Engineering

1.7.1 Text Preprocessing

[9]:

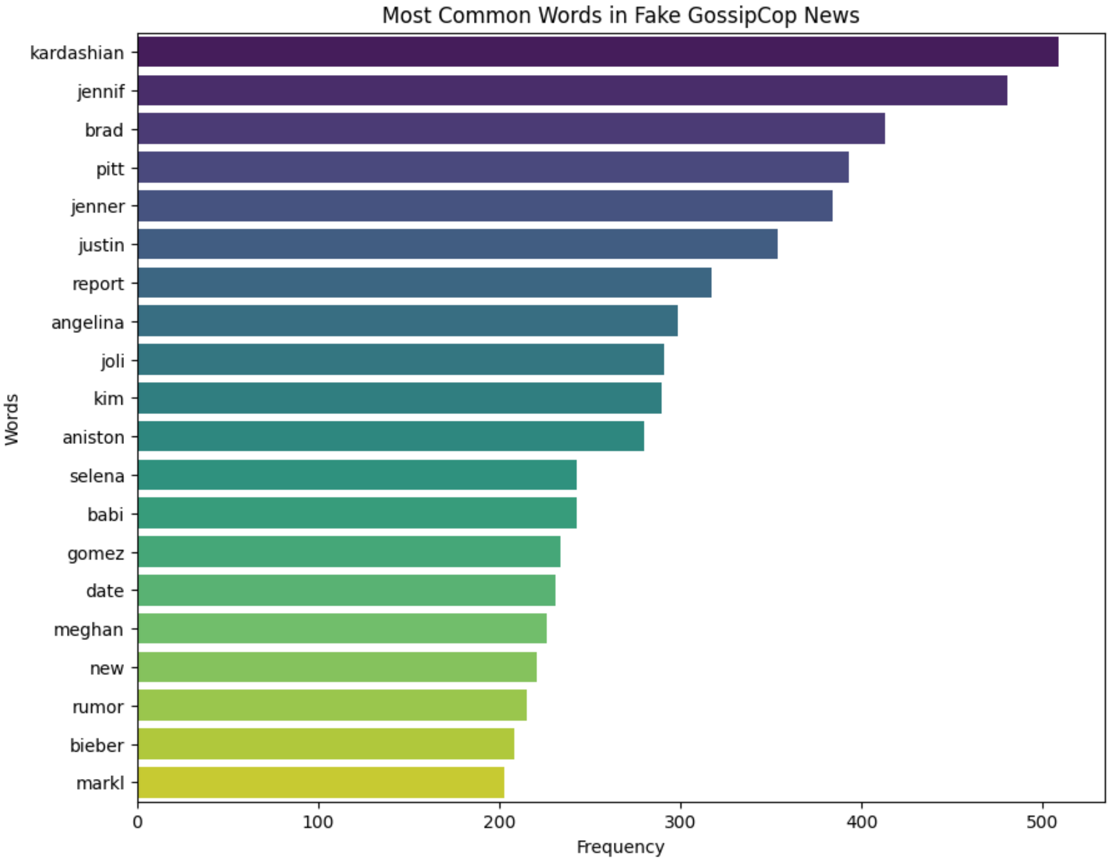
```python
# Initialize stopwords and stemmer
stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()

def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()
    # Remove punctuation and non-word characters
    text = re.sub(r'\W+', ' ', text)
    # Stemming and stopword removal
    words = text.split()
    filtered_words = [stemmer.stem(word) for word in words if word not in␣
 ↪stop_words]
    return ' '.join(filtered_words)
```

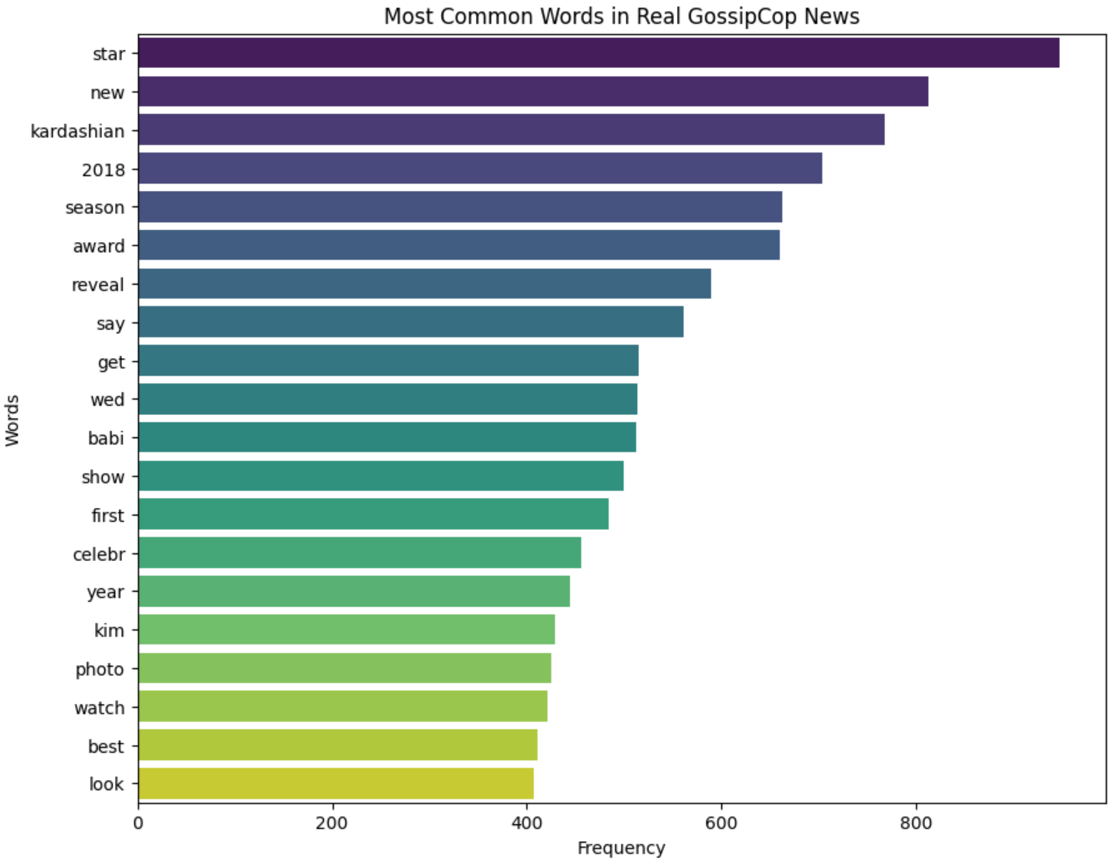
```
# Apply preprocessing to each title
gossipcop_fake['title'] = gossipcop_fake['title'].apply(preprocess_text)
gossipcop_real['title'] = gossipcop_real['title'].apply(preprocess_text)
politifact_fake['title'] = politifact_fake['title'].apply(preprocess_text)
politifact_real['title'] = politifact_real['title'].apply(preprocess_text)
```

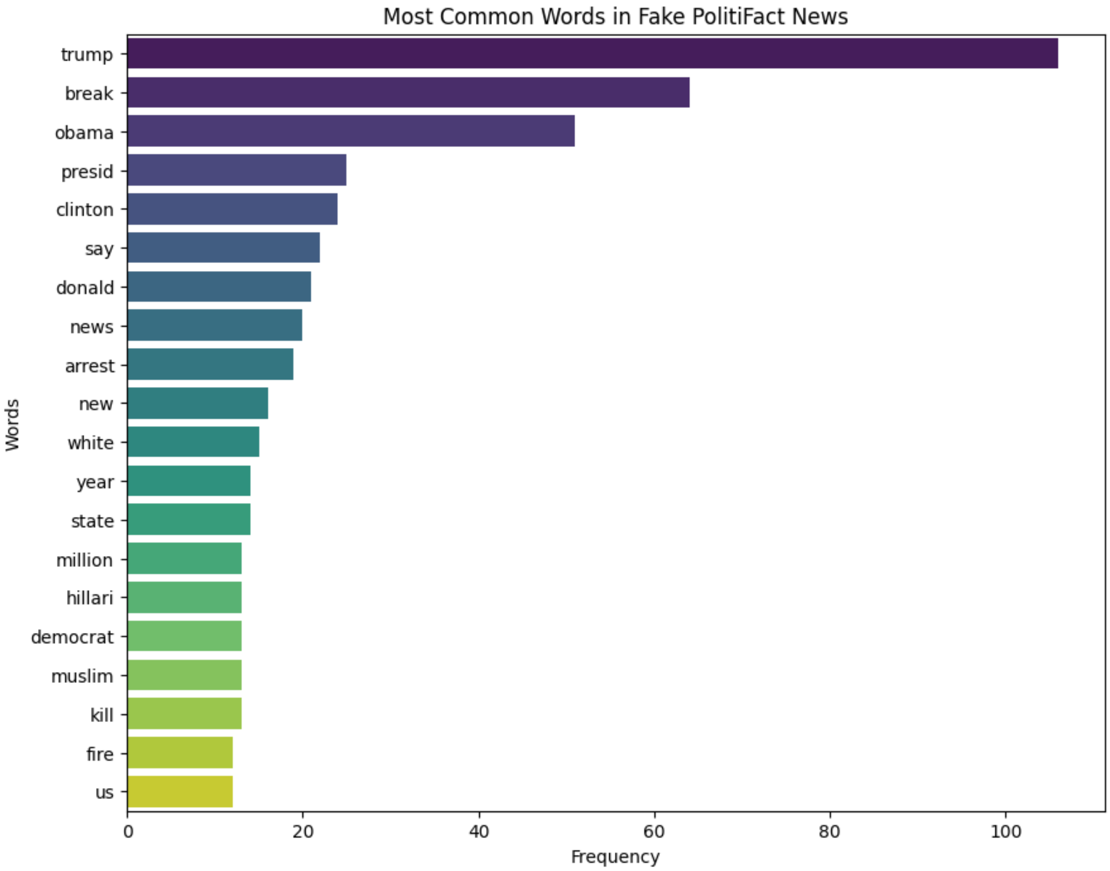
1.7.2 Visualize the Most Common Words After Preprocessing

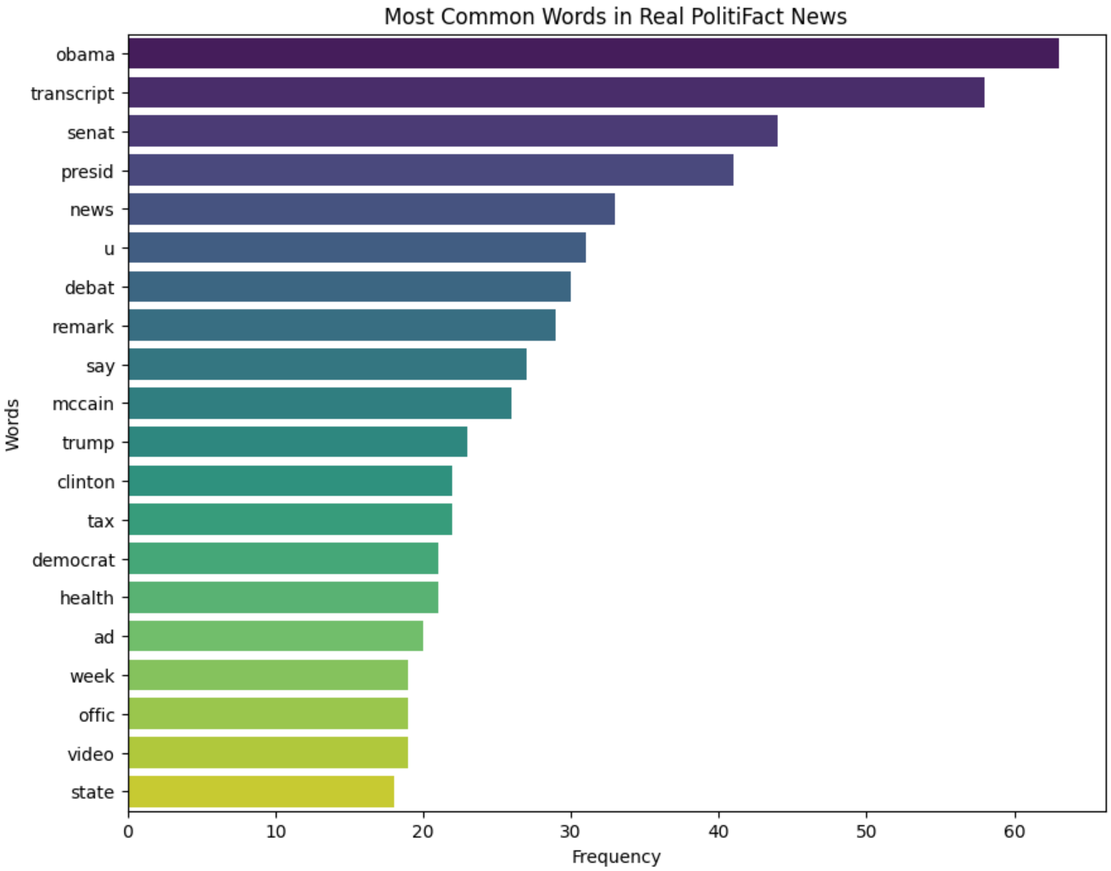
[10]:

```
# Function to plot most common words
def plot_most_common_words(preprocessed_texts, title, num=20):
    all_words = ' '.join(preprocessed_texts).split()
    freq_dist = Counter(all_words)
    common_words = freq_dist.most_common(num)

    words = [word[0] for word in common_words]
    counts = [word[1] for word in common_words]
    data = pd.DataFrame({'Word': words, 'Frequency': counts})

    plt.figure(figsize=(10, 8))
    sns.barplot(data=data, y='Word', x='Frequency', hue='Word', dodge=False,␣
  ↪palette="viridis")
    plt.legend([],[], frameon=False) # Hides the legend as it's redundant in␣
  ↪this context
    plt.title(f'Most Common Words in {title}')
    plt.xlabel('Frequency')
    plt.ylabel('Words')
    plt.show()

# Apply the function to visualize data
plot_most_common_words(gossipcop_fake['title'], 'Fake GossipCop News')
plot_most_common_words(gossipcop_real['title'], 'Real GossipCop News')
plot_most_common_words(politifact_fake['title'], 'Fake PolitiFact News')
plot_most_common_words(politifact_real['title'], 'Real PolitiFact News')
```


Most Common Words in Fake GossipCop News
Words
kardashian
jennif
brad
pitt
jenner
justin
report
angelina
joli
kim
aniston
selena
babi
gomez
date
meghan
new
rumor
bieber
markl
0
100
200
300
400
500
Frequency


Most Common Words in Real GossipCop News
Words
star
new
kardashian
2018
season
award
reveal
say
get
wed
babi
show
first
celebr
year
kim
photo
watch
best
look
0
200
400
600
800
Frequency


Most Common Words in Fake PolitiFact News
Words
trump
break
obama
presid
clinton
say
donald
news
arrest
new
white
year
state
million
hillari
democrat
muslim
kill
fire
us
0
20
40
60
80
100
Frequency



The most common words in fake news GossipCop are: kardashian, jennif, brad. The most common words in real news GossipCop are: star, new, kardashian. The most common words in fake news PolitiFact are: trump, break, obama. The most common words in real news PolitiFact are: obama, transcript, senat.

1.8 Combine and Shuffle the Datasets

[11]:

```
# Combine fake and real datasets for GossipCop and shuffle
gossipcop_combined = pd.concat([gossipcop_fake, gossipcop_real]).
 ↪reset_index(drop=True)
gossipcop_combined = shuffle(gossipcop_combined, random_state=42)

# Combine fake and real datasets for PolitiFact and shuffle
politifact_combined = pd.concat([politifact_fake, politifact_real]).
 ↪reset_index(drop=True)
politifact_combined = shuffle(politifact_combined, random_state=42)

# Check the combined datasets
print(gossipcop_combined.head())
print(politifact_combined.head())
```

```
                                                   title  label
11080  lea michel hairstylist mix textur spray coconu…      1
291    thoma markl princ harri polit miss daughter me…      0
17231  2019 sag award nomin see full list nomine varieti      1
16382  see meghan markl royal coat arm symbol hide wi…      1
9364     kyli jenner visit shaman life kyli season final      1
                                                  title  label
260          world popular candi remov shelv octob 2017      0
832                              brows congression bill      1
846                           suprem court vacanc video      1
1007                                    u import export      1
88    die 78 year old cia agent admit kill marilyn m…      0
```

1.9 Save Preprocessed datasets to Google drive

[12]:
```python
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Save preprocessed data to Google Drive
gossipcop_combined.to_csv('/content/drive/My Drive/gossipcop_preprocessed.csv',␣
  ↪index=False)
politifact_combined.to_csv('/content/drive/My Drive/politifact_preprocessed.
  ↪csv', index=False)
```

```
Mounted at /content/drive
```

1.10 References

[7] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. https://doi.org/10.48550/arXiv.1809.01286

[8] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context, and Dynamic Information for Studying Fake News on Social Media. https://github.com/KaiDMML/FakeNewsNet

[ ]:
```python
# Install LaTeX packages necessary for converting notebooks to PDF
!apt-get update
!apt-get install -y texlive-xetex texlive-fonts-recommended␣
  ↪texlive-plain-generic texlive-latex-extra pandoc

# Convert the notebook to PDF
!jupyter nbconvert --to pdf "/content/drive/My Drive/Colab Notebooks/
  ↪FakeNewsNetPreprocessing.ipynb"
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,626 B]
Hit:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
```

```
InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Get:6 https://r2u.stat.illinois.edu/ubuntu jammy Release [5,713 B]
Get:7 https://r2u.stat.illinois.edu/ubuntu jammy Release.gpg [793 B]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:9 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Get:10 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease [24.3 kB]
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:13 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,544 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
[1,127 kB]
Get:15 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,181 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,263
kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages
[1,998 kB]
Get:18 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy/main
amd64 Packages [48.1 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[1,410 kB]
Fetched 17.9 MB in 2s (7,257 kB/s)
Reading package lists… Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc-data
poppler-data
  preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick ruby-
xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base texlive-
binaries
  texlive-latex-base texlive-latex-recommended texlive-pictures tipa xfonts-
```

```
encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java texlive-
luatex
  pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc nodejs php python
libjs-mathjax
  libjs-katex citation-style-language-styles poppler-utils ghostscript fonts-
japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-
arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv | postscript-
viewer perl-tk xpdf
  | pdf-viewer xzdec texlive-fonts-recommended-doc texlive-latex-base-doc
python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-
extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex texlive-
pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc
pandoc-data
  poppler-data preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-
webrick ruby-xmlrpc
  ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
texlive-binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-extra texlive-
latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa xfonts-encodings
xfonts-utils
0 upgraded, 58 newly installed, 0 to remove and 45 not upgraded.
Need to get 202 MB of archives.
After this operation, 728 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
```

```
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.7 [752 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.7 [5,028 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-
gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-gfm-
extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.7 [50.1 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
```

```
Get:28 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ruby-webrick all
1.7.0-3 [51.8 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.7 [5,113 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc-data all
2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc amd64
2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
```

```
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 7s (27.1 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 123576 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.7_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.7) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.7_amd64.deb …
```

```
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.7) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack …/17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack …/18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/19-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/20-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/21-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/23-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
```

```
Preparing to unpack …/24-ruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/25-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/26-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/27-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/28-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/29-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/31-libruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/32-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/33-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/36-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/38-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
```

```
Preparing to unpack …/39-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package pandoc-data.
Preparing to unpack …/40-pandoc-data_2.9.2.1-3ubuntu2_all.deb …
Unpacking pandoc-data (2.9.2.1-3ubuntu2) …
Selecting previously unselected package pandoc.
Preparing to unpack …/41-pandoc_2.9.2.1-3ubuntu2_amd64.deb …
Unpacking pandoc (2.9.2.1-3ubuntu2) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/42-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/43-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/44-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/45-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/47-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/48-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/49-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/50-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/51-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/52-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/53-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
```

FakeNewsNetClassifier

July 14, 2024

1 FakeNewsNetClassifier

This notebook builds and evaluates machine learning models for detecting fake news using the FakeNewsNet dataset, that contians two subsets, GossipCop and PolitiFact. I will be using three different classifiers: Logistic Regression, Support Vector Machine (SVM), and Naive Bayes. I will also balance the dataset using SMOTE and perform hyperparameter tuning to improve models' performance

1.1 Import necessary libraries

[1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score,␣
 ↪GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score, confusion_matrix, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from imblearn.over_sampling import SMOTE
import joblib
```

1.2 Load Preprocessed Data from Google Drive

[2]:

```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Load the preprocessed datasets from Google Drive
gossipcop_combined = pd.read_csv('/content/drive/My Drive/
 ↪gossipcop_preprocessed.csv')
politifact_combined = pd.read_csv('/content/drive/My Drive/
 ↪politifact_preprocessed.csv')
```

```
# Quick check of the data loaded
print(gossipcop_combined.head())
print(politifact_combined.head())
```

```
Mounted at /content/drive
                                               title  label
0  lea michel hairstylist mix textur spray coconu…      1
1  thoma markl princ harri polit miss daughter me…      0
2  2019 sag award nomin see full list nomine varieti      1
3  see meghan markl royal coat arm symbol hide wi…      1
4    kyli jenner visit shaman life kyli season final      1
                                               title  label
0         world popular candi remov shelv octob 2017      0
1                              brows congression bill      1
2                          suprem court vacanc video      1
3                                    u import export      1
4  die 78 year old cia agent admit kill marilyn m…      0
```

1.3 Check loaded data

[3]:

```
# Check for NaN values in datasets
print("NaN values in GossipCop dataset:\n", gossipcop_combined.isnull().sum())
print("NaN values in PolitiFact dataset:\n", politifact_combined.isnull().sum())

# Ensure there are no NaN values
gossipcop_combined.dropna(inplace=True)
politifact_combined.dropna(inplace=True)

# Re-check for NaN values in datasets
print("NaN values in GossipCop dataset after dropna:\n", gossipcop_combined.
 ↪isnull().sum())
print("NaN values in PolitiFact dataset after dropna:\n", politifact_combined.
 ↪isnull().sum())
```

```
NaN values in GossipCop dataset:
 title    1
label    0
dtype: int64
NaN values in PolitiFact dataset:
 title    2
label    0
dtype: int64
NaN values in GossipCop dataset after dropna:
 title    0
label    0
dtype: int64
NaN values in PolitiFact dataset after dropna:
 title    0
```

```
label    0
dtype: int64
```

1.4 Split Data into Training and Testing Sets

I am splitting the dataset the same way as Shu et al. [7]: “We use 80% of data for training and 20% for testing.”

[4]:
```
# Define features (X) and labels (y)
X_gossipcop = gossipcop_combined['title']
y_gossipcop = gossipcop_combined['label']

X_politifact = politifact_combined['title']
y_politifact = politifact_combined['label']

# Split the data into training and testing sets (80% train, 20% test)
X_train_gossipcop, X_test_gossipcop, y_train_gossipcop, y_test_gossipcop =␣
 ↪train_test_split(X_gossipcop, y_gossipcop, test_size=0.2, random_state=42)
X_train_politifact, X_test_politifact, y_train_politifact, y_test_politifact =␣
 ↪train_test_split(X_politifact, y_politifact, test_size=0.2, random_state=42)
```

1.5 Vectorize the Text Data Using TF-IDF

Shen et al [6] applied TF-IDF technique for feature extraction, so I will use this too. TF-IDF should be applied after splitting the data to avoid data leakage. The vectorizer is only fitted on the training data and then applied to the test data.

[5]:
```
# Initialize the TF-IDF Vectorizer
vectorizer = TfidfVectorizer(max_features=1000)

# Fit and transform the training data, transform the test data
X_train_gossipcop_tfidf = vectorizer.fit_transform(X_train_gossipcop)
X_test_gossipcop_tfidf = vectorizer.transform(X_test_gossipcop)

X_train_politifact_tfidf = vectorizer.fit_transform(X_train_politifact)
X_test_politifact_tfidf = vectorizer.transform(X_test_politifact)
```

1.6 Balance classes using SMOTE

SMOTE is a popular technique, that can be applied to balance an imbalanced dataset as descibed by Chawla et al.[14]. It generates synthetic examples for the minority class. The method combines minority class examples to create synthetic examples and balances the class distribution. It improves classifier performance on imbalanced datasets. I apply this technique to balance the class imbalance present in the original datasets.

[6]:
```
# Apply SMOTE to balance the classes in the training data
smote = SMOTE(random_state=42)
```

```
X_train_gossipcop_resampled, y_train_gossipcop_resampled = smote.
  ↪fit_resample(X_train_gossipcop_tfidf, y_train_gossipcop)
X_train_politifact_resampled, y_train_politifact_resampled = smote.
  ↪fit_resample(X_train_politifact_tfidf, y_train_politifact)

# Check class distribution after SMOTE
print("Class distribution in resampled GossipCop training set:")
print(pd.Series(y_train_gossipcop_resampled).value_counts())

print("Class distribution in resampled PolitiFact training set:")
print(pd.Series(y_train_politifact_resampled).value_counts())
```

```
Class distribution in resampled GossipCop training set:
label
0    13428
1    13428
Name: count, dtype: int64
Class distribution in resampled PolitiFact training set:
label
0    492
1    492
Name: count, dtype: int64
```

1.7 Train and Evaluate Classifiers

[7]:

```
# Train and evaluate the model, returning the model and predictions
def train_and_evaluate(model, X_train, X_test, y_train, y_test):
    # Train the model
    model.fit(X_train, y_train)
    # Predict on the test data
    y_pred = model.predict(X_test)
    # Calculate evaluation metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    roc_auc = roc_auc_score(y_test, y_pred)
    # Print evaluation metrics
    print(f'Accuracy: {accuracy:.4f}')
    print(f'Precision: {precision:.4f}')
    print(f'Recall: {recall:.4f}')
    print(f'F1-score: {f1:.4f}')
    print(f'ROC AUC: {roc_auc:.4f}')
    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    print('Confusion Matrix:')
    print(cm)
```

```
    # Plot Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['False',␣
  ↪'True'], yticklabels=['False', 'True'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()
    # Plot ROC Curve
    if hasattr(model, "predict_proba"):
        y_proba = model.predict_proba(X_test)[:, 1]
    else:  # Use decision_function if predict_proba is not available
        y_proba = model.decision_function(X_test)
        y_proba = (y_proba - y_proba.min()) / (y_proba.max() - y_proba.min())
    fpr, tpr, thresholds = roc_curve(y_test, y_proba)
    plt.figure(figsize=(8, 6))
    plt.plot(fpr, tpr, label='ROC Curve (area = %0.2f)' % roc_auc)
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic')
    plt.legend(loc="lower right")
    plt.show()
    # Return the trained model and predictions
    return model, y_pred
```

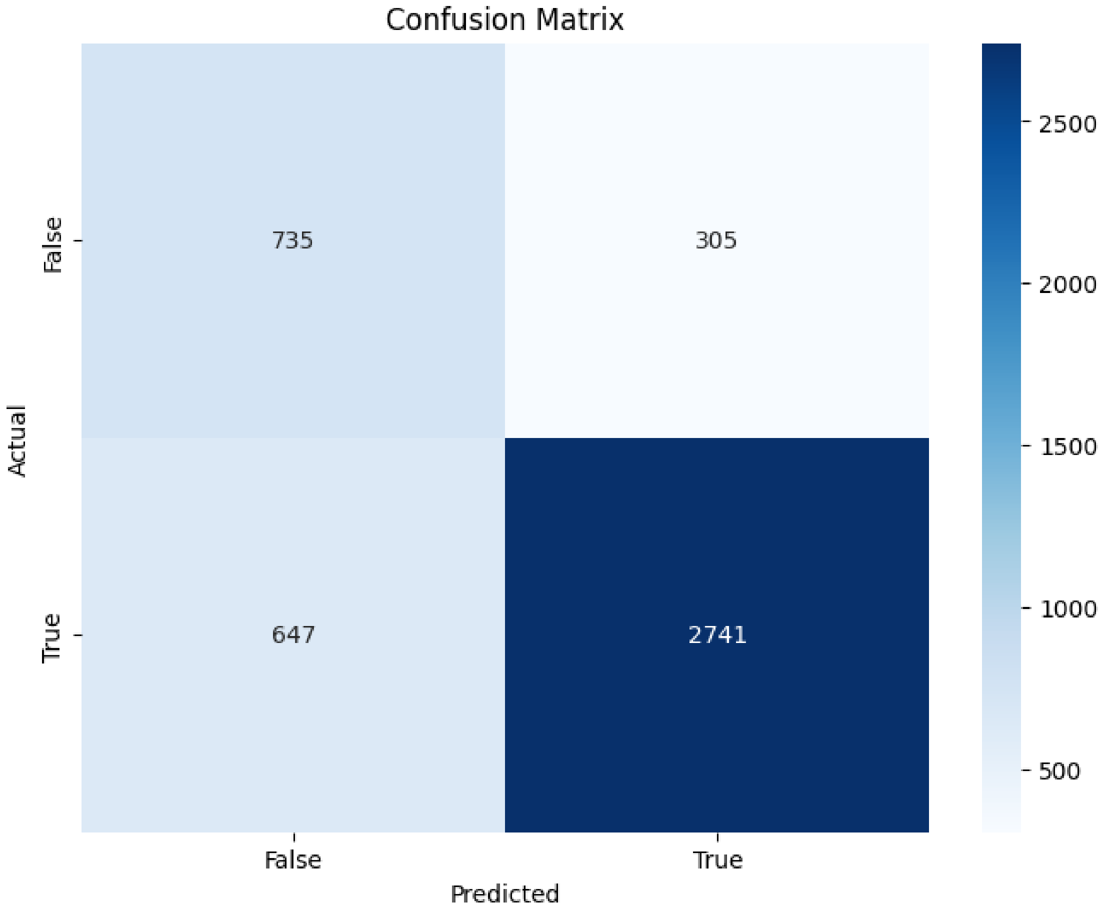
1.7.1 Model Training and Evaluation

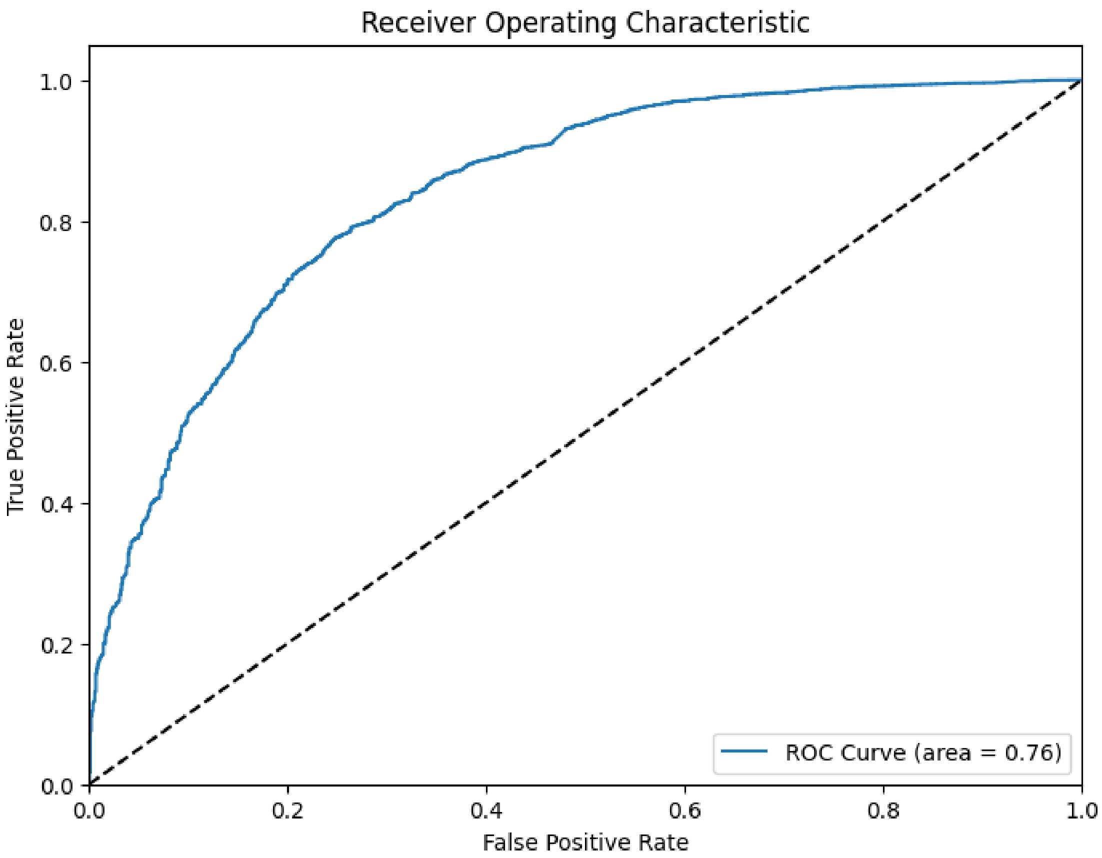
[8]:

```
# Logistic Regression
print("Logistic Regression (GossipCop):")
logistic_model_gossipcop = LogisticRegression(max_iter=1000)
train_and_evaluate(logistic_model_gossipcop, X_train_gossipcop_resampled,␣
  ↪X_test_gossipcop_tfidf, y_train_gossipcop_resampled, y_test_gossipcop)

print("\nLogistic Regression (PolitiFact):")
logistic_model_politifact = LogisticRegression(max_iter=1000)
train_and_evaluate(logistic_model_politifact, X_train_politifact_resampled,␣
  ↪X_test_politifact_tfidf, y_train_politifact_resampled, y_test_politifact)

# Support Vector Machine (SVM)
print("\nSupport Vector Machine (GossipCop):")
svm_model_gossipcop = SVC(kernel='linear')
```

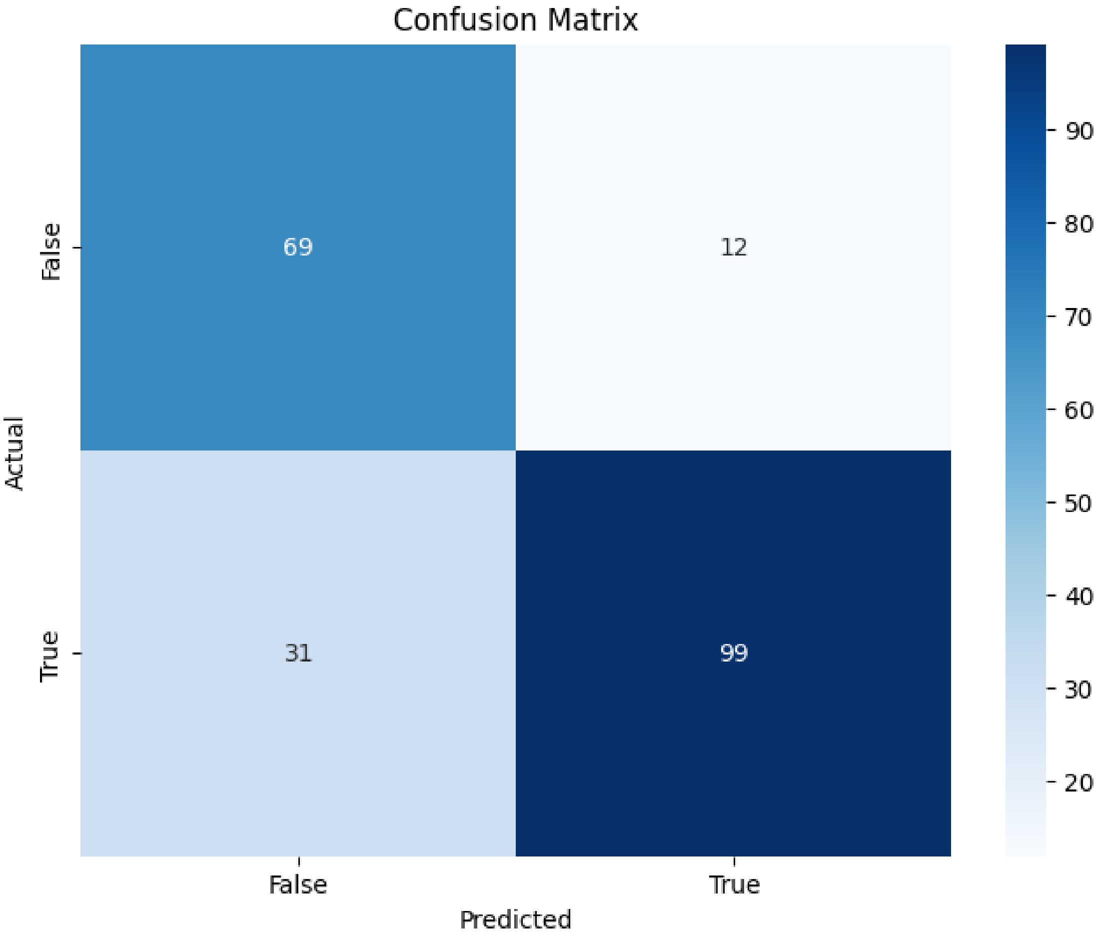
```
train_and_evaluate(svm_model_gossipcop, X_train_gossipcop_resampled,␣
 ↪X_test_gossipcop_tfidf, y_train_gossipcop_resampled, y_test_gossipcop)

print("\nSupport Vector Machine (PolitiFact):")
svm_model_politifact = SVC(kernel='linear')
train_and_evaluate(svm_model_politifact, X_train_politifact_resampled,␣
 ↪X_test_politifact_tfidf, y_train_politifact_resampled, y_test_politifact)

# Naive Bayes
print("\nNaive Bayes (GossipCop):")
nb_model_gossipcop = MultinomialNB()
train_and_evaluate(nb_model_gossipcop, X_train_gossipcop_resampled,␣
 ↪X_test_gossipcop_tfidf, y_train_gossipcop_resampled, y_test_gossipcop)

print("\nNaive Bayes (PolitiFact):")
nb_model_politifact = MultinomialNB()
train_and_evaluate(nb_model_politifact, X_train_politifact_resampled,␣
 ↪X_test_politifact_tfidf, y_train_politifact_resampled, y_test_politifact)
```

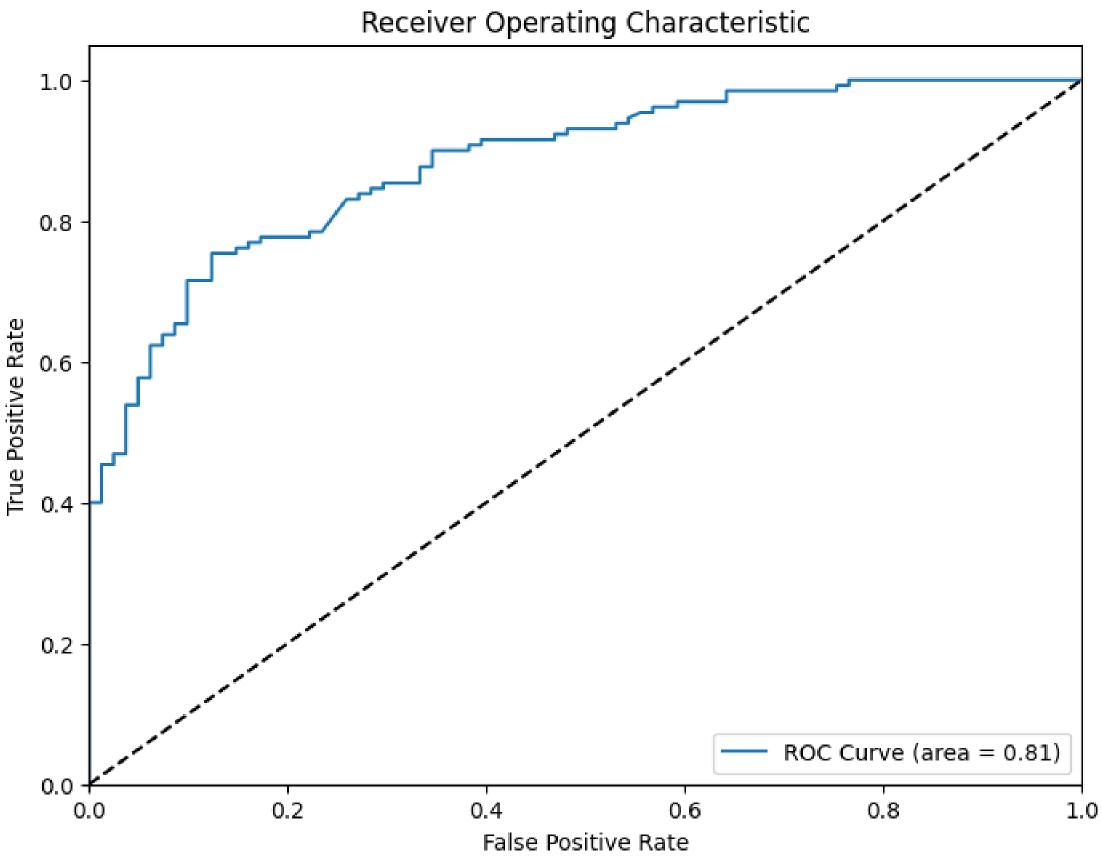
```
Logistic Regression (GossipCop):
Accuracy: 0.7850
Precision: 0.8999
Recall: 0.8090
F1-score: 0.8520
ROC AUC: 0.7579
Confusion Matrix:
[[ 735  305]
 [ 647 2741]]
```


Confusion Matrix
735
305
647
2741
False
True
Actual
False
True
Predicted
2500
2000
1500
1000
500

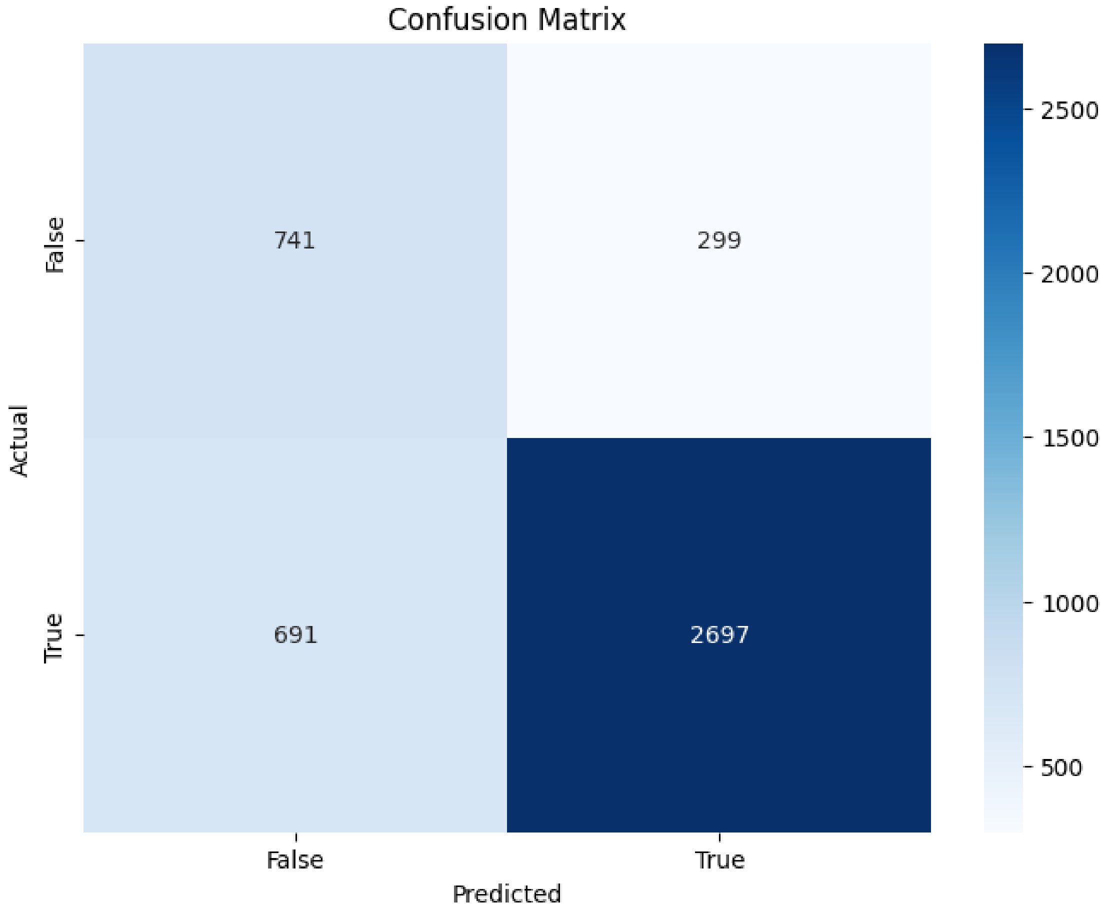


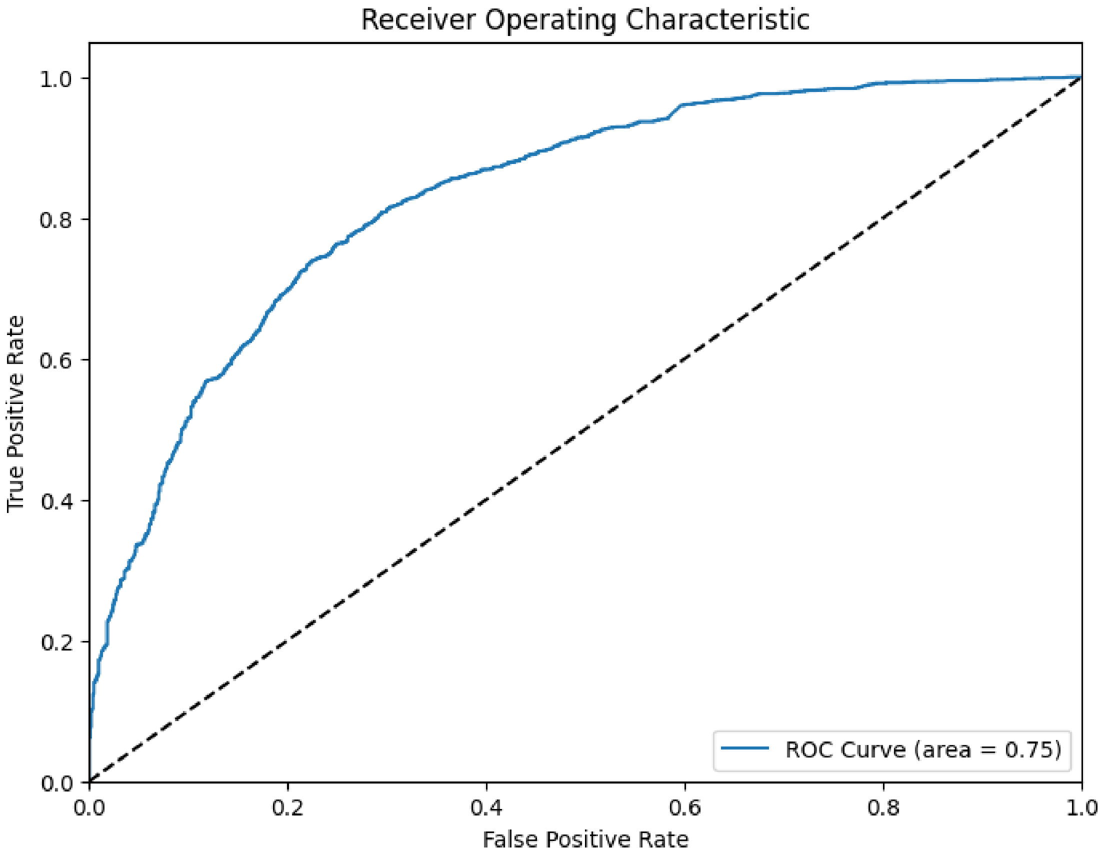
```
Logistic Regression (PolitiFact):
Accuracy: 0.7962
Precision: 0.8919
Recall: 0.7615
F1-score: 0.8216
ROC AUC: 0.8067
Confusion Matrix:
[[69 12]
 [31 99]]
```


Confusion Matrix
Actual
False
True
Predicted
False
True
69
12
31
99
90
80
70
60
50
40
30
20

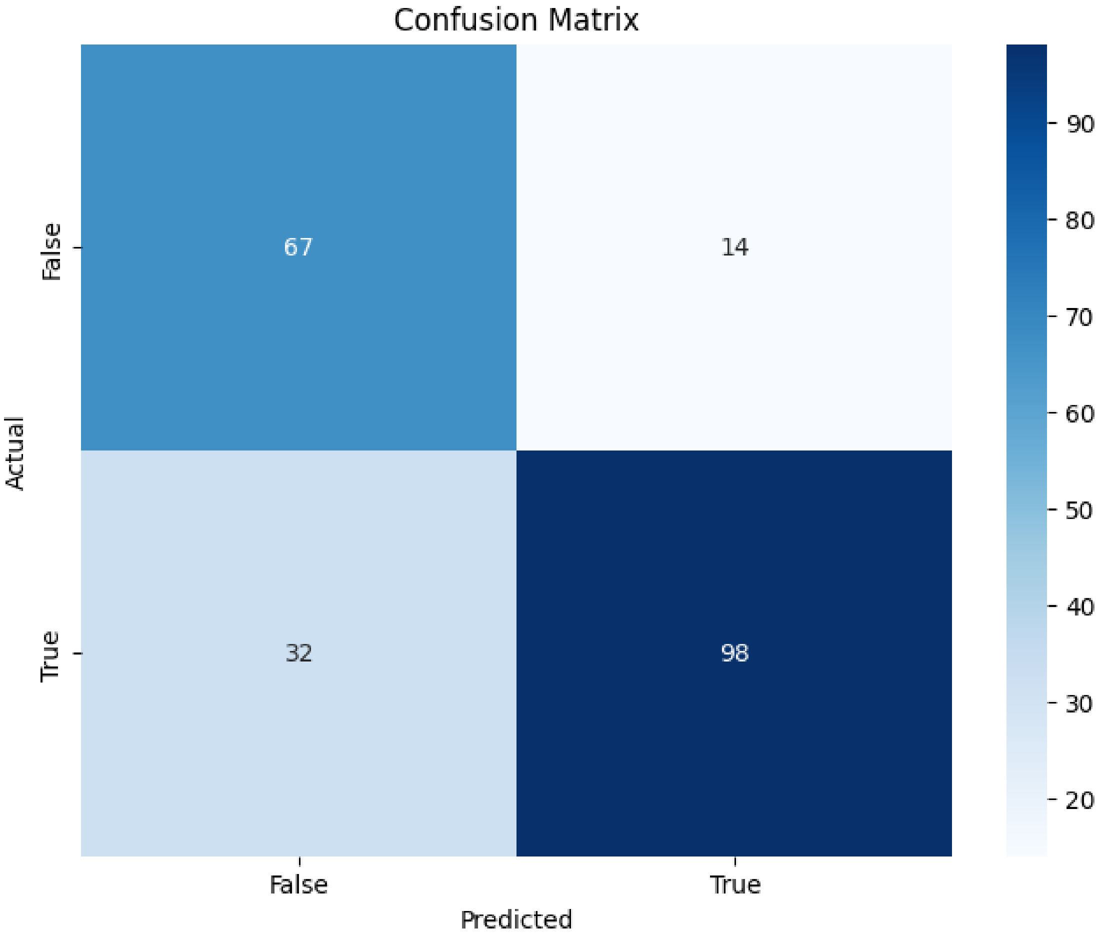


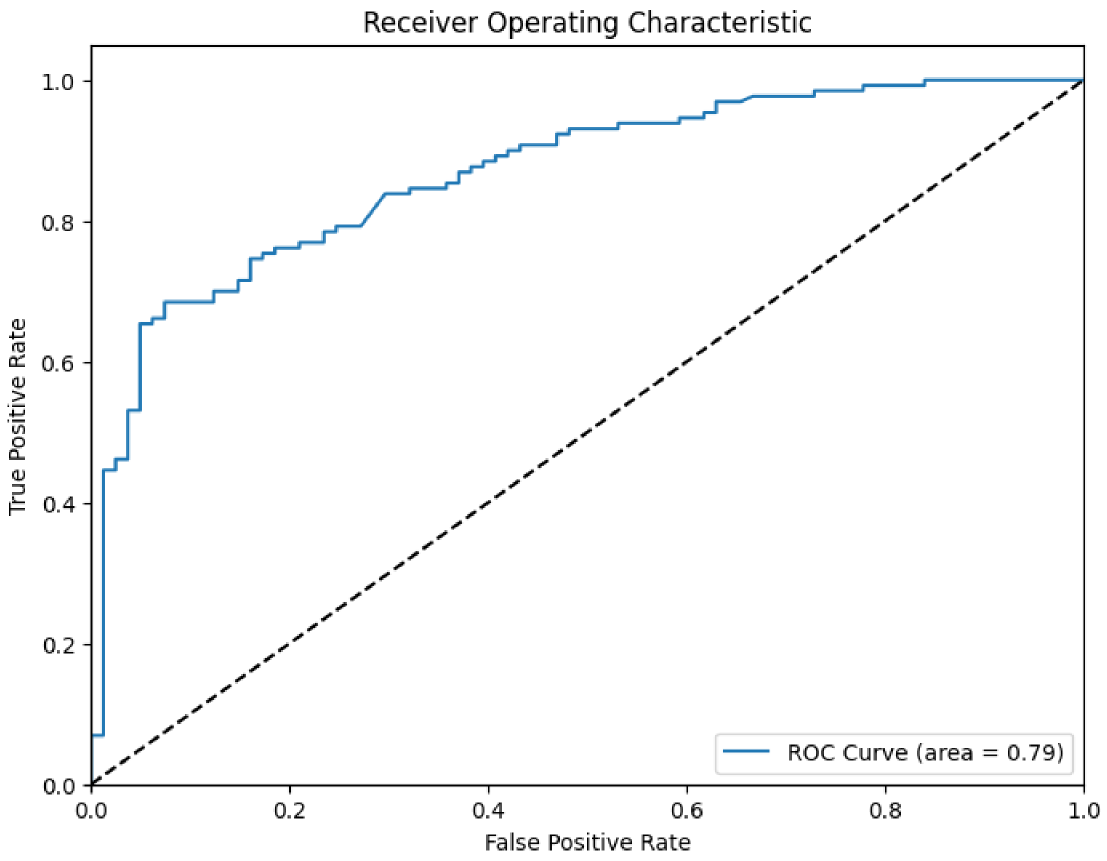
```
Support Vector Machine (GossipCop):
Accuracy: 0.7764
Precision: 0.9002
Recall: 0.7960
F1-score: 0.8449
ROC AUC: 0.7543
Confusion Matrix:
[[ 741  299]
 [ 691 2697]]
```


Confusion Matrix
741
299
691
2697
False
True
Actual
False
True
Predicted
2500
2000
1500
1000
500

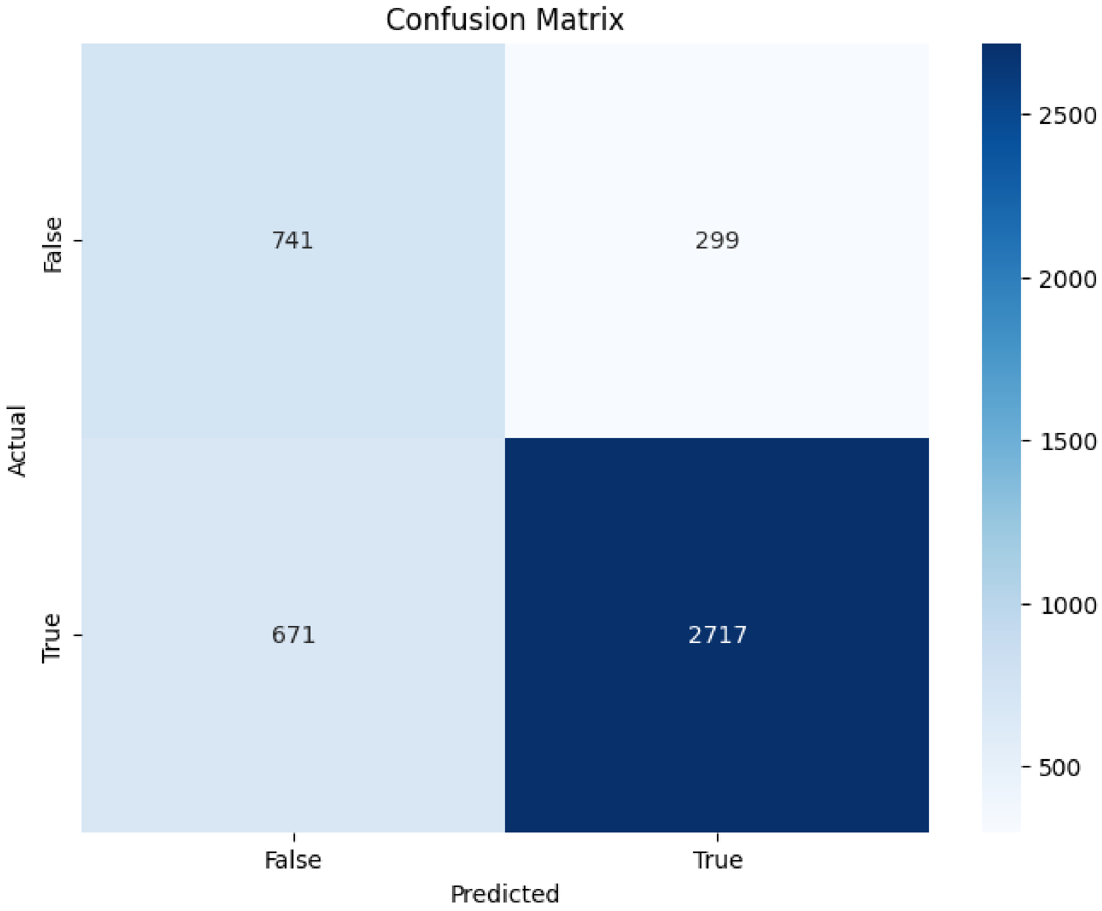


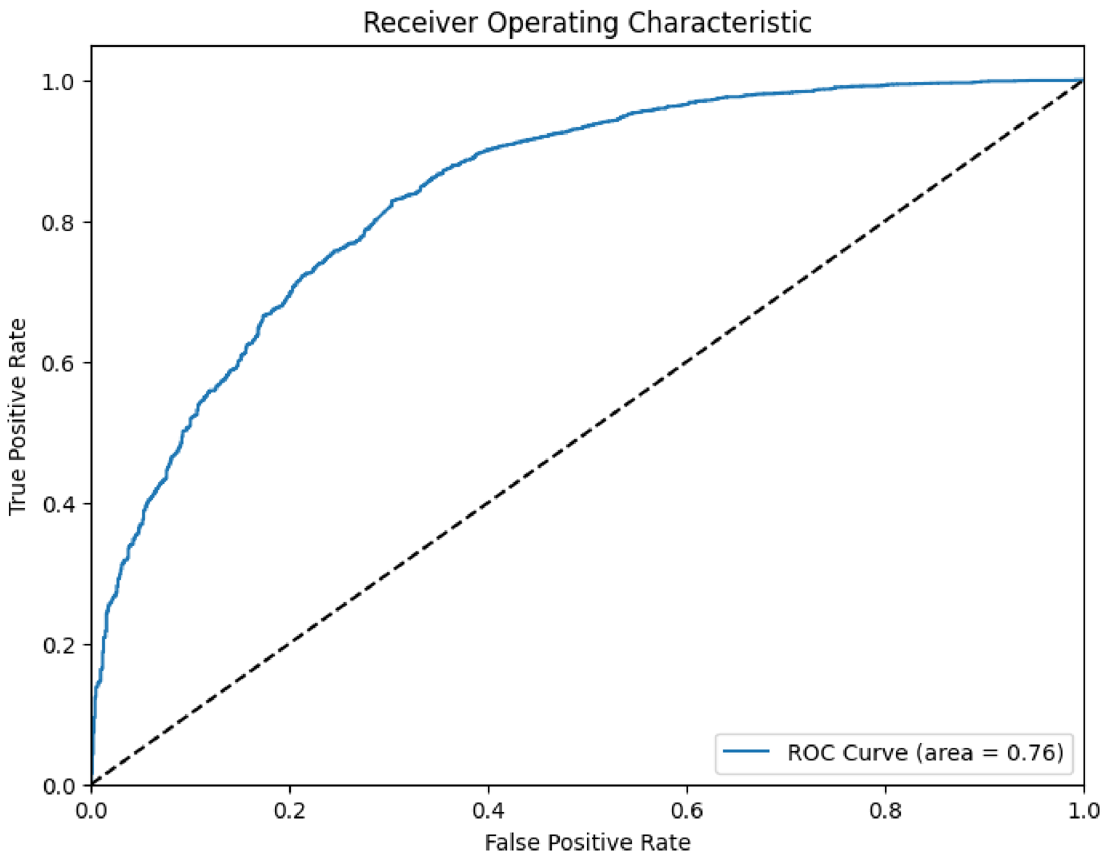
```
Support Vector Machine (PolitiFact):
Accuracy: 0.7820
Precision: 0.8750
Recall: 0.7538
F1-score: 0.8099
ROC AUC: 0.7905
Confusion Matrix:
[[67 14]
 [32 98]]
```


Confusion Matrix
Actual
False
True
Predicted
False
True
67
14
32
98
90
80
70
60
50
40
30
20

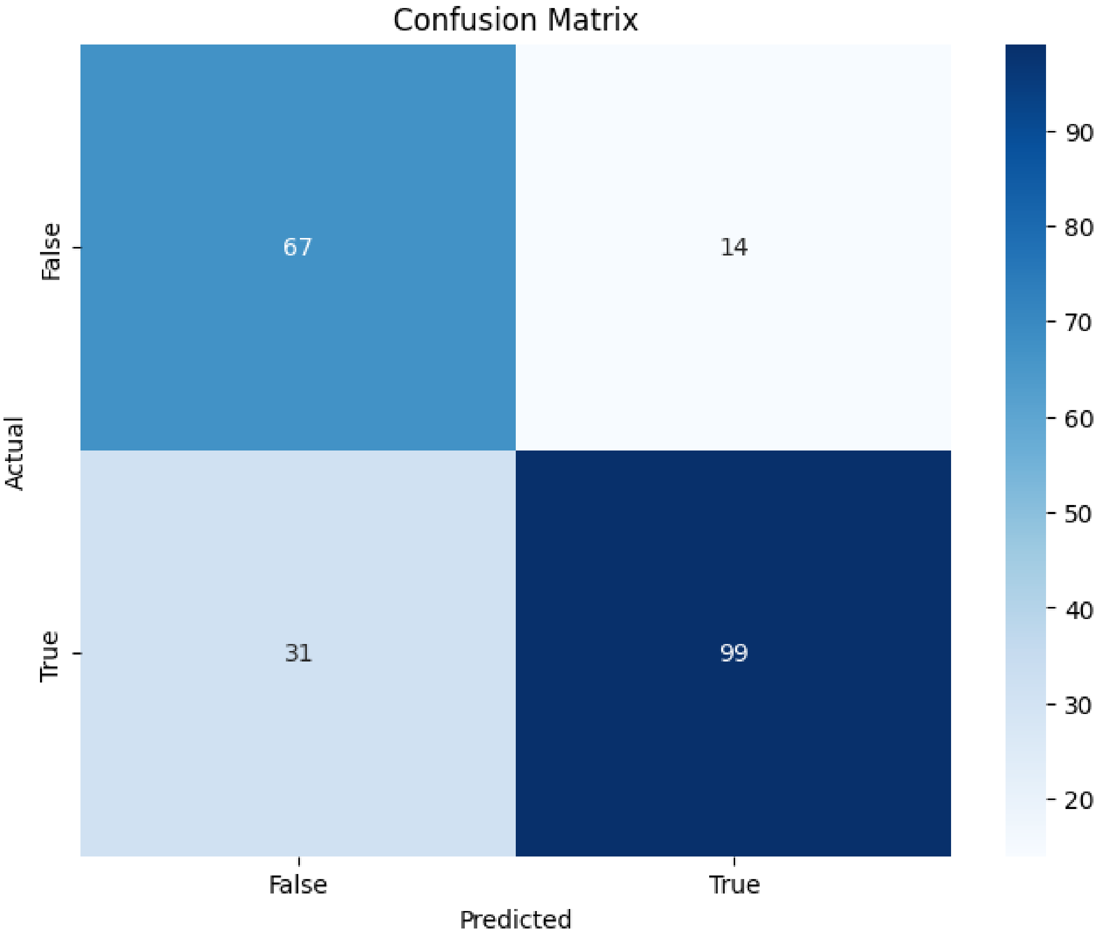


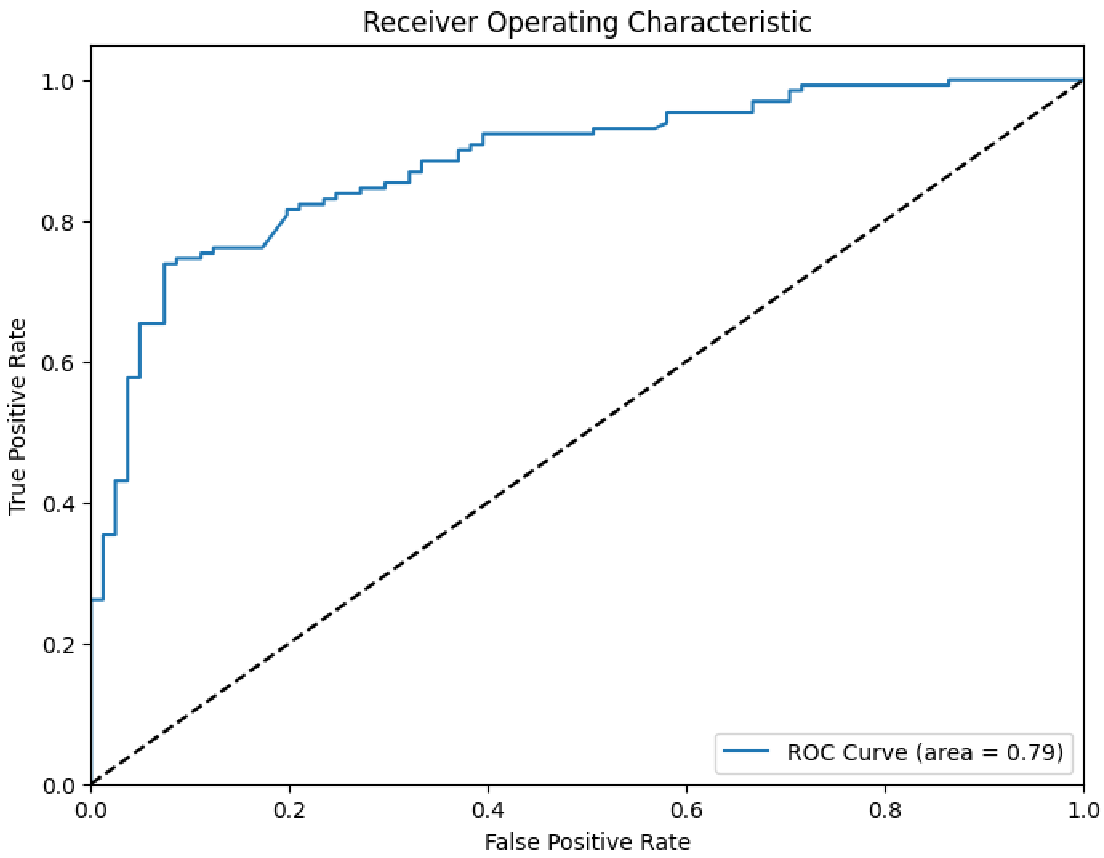
```
Naive Bayes (GossipCop):
Accuracy: 0.7809
Precision: 0.9009
Recall: 0.8019
F1-score: 0.8485
ROC AUC: 0.7572
Confusion Matrix:
[[ 741  299]
 [ 671 2717]]
```


Confusion Matrix
Actual
False
True
741
299
671
2717
Predicted
False
True
2500
2000
1500
1000
500



```
Naive Bayes (PolitiFact):
Accuracy: 0.7867
Precision: 0.8761
Recall: 0.7615
F1-score: 0.8148
ROC AUC: 0.7943
Confusion Matrix:
[[67 14]
 [31 99]]
```


Confusion Matrix
67
14
31
99
False
True
Actual
False
True
Predicted
90
80
70
60
50
40
30
20



```
[8]: (MultinomialNB(),
      array([1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1,
             0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0,
             0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
             1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0,
             1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
             1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1,
             0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0,
             1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1,
             0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
             0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]))
```

1.7.2 Comparison of my results with baseline Shu et al. [7]

GossipCop Dataset

Support Vector Machine (SVM)

```
My Results:
    Accuracy: 0.7764
    Precision: 0.9002
```

```
    Recall: 0.7960
    F1-score: 0.8449
Baseline Results:
    Accuracy: 0.497
    Precision: 0.511
    Recall: 0.713
    F1-score: 0.595
```

Logistic Regression

```
My Results:
    Accuracy: 0.7850
    Precision: 0.8999
    Recall: 0.8090
    F1-score: 0.8520
Baseline Results:
    Accuracy: 0.648
    Precision: 0.675
    Recall: 0.619
    F1-score: 0.646
```

Naive Bayes

```
My Results:
    Accuracy: 0.7809
    Precision: 0.9009
    Recall: 0.8019
    F1-score: 0.8485
Baseline Results:
    Accuracy: 0.624
    Precision: 0.631
    Recall: 0.669
    F1-score: 0.649
```

PolitiFact Dataset

Support Vector Machine (SVM)

```
My Results:
    Accuracy: 0.7820
    Precision: 0.8750
    Recall: 0.7538
    F1-score: 0.8099
Baseline Results:
    Accuracy: 0.580
    Precision: 0.611
    Recall: 0.717
    F1-score: 0.659
```

Logistic Regression

```
My Results:
```

```
    Accuracy: 0.7962
    Precision: 0.8919
    Recall: 0.7615
    F1-score: 0.8216
Baseline Results:
    Accuracy: 0.642
    Precision: 0.757
    Recall: 0.543
    F1-score: 0.633
```

Naive Bayes

```
My Results:
    Accuracy: 0.7867
    Precision: 0.8761
    Recall: 0.7615
    F1-score: 0.8148
Baseline Results:
    Accuracy: 0.617
    Precision: 0.674
    Recall: 0.630
    F1-score: 0.651
```

My models outperform the baseline results significantly across all metrics (accuracy, precision, recall, F1-score) for both datasets (GossipCop and PolitiFact). After applying SMOTE to balance the datasets, the models show a more balanced performance across precision and recall, which results in higher F1-scores.

1.8 Implement 5-Fold Cross-Validation

I will randomly split the datasets into five parts and conduct 5-fold cross-validation to obtain robust results, as Bian et al.[5] did. Cross-validation is a statistical method used to estimate the performance of machine learning models. As Browne et al.[15] explain: "In its simplest form, the leaving one out at a time method, this involves partitioning a sample of size N into a calibration sample of size N-1 and a validation sample of size 1 and repeating the process N times. An average of the N cross-validation index values is then used." This method involves splitting the data into a number of subsets (folds), training the model on some subsets while testing it on the remaining subset, and repeating this process several times. The performance metrics are then averaged over all iterations to provide a more robust evaluation.

According to Powers et al.[16], "The F1-score, which is the harmonic mean of precision and recall, is particularly recommended for imbalanced datasets because it provides a balance between false positives and false negatives, thus giving a more comprehensive measure of a model's performance." For this reason, I have selected the F1 score as the scoring parameter in 5-Fold Cross-Validation, because the datasets are imbalanced.

1.8.1 Cross-Validation Function

[9]:

```
# Function to perform cross-validation
def cross_validate_model(model, X, y, cv=5, scoring='f1'):
    # Perform cross-validation
    scores = cross_val_score(model, X, y, cv=cv, scoring=scoring)
    print(f'Cross-Validation Scores: {scores}')
    print(f'Average Cross-Validation Score: {np.mean(scores):.4f}')
```

1.8.2 Cross-Validation Scores

[10]:

```
# Logistic Regression
print("Logistic Regression (Cross-Validation) (GossipCop):")
logistic_model_gossipcop = LogisticRegression(max_iter=1000)
cross_validate_model(logistic_model_gossipcop, X_train_gossipcop_resampled,␣
 ↪y_train_gossipcop_resampled)

print("\nLogistic Regression (Cross-Validation) (PolitiFact):")
logistic_model_politifact = LogisticRegression(max_iter=1000)
cross_validate_model(logistic_model_politifact, X_train_politifact_resampled,␣
 ↪y_train_politifact_resampled)

# Support Vector Machine (SVM)
print("\nSupport Vector Machine (Cross-Validation) (GossipCop):")
svm_model_gossipcop = SVC(kernel='linear')
cross_validate_model(svm_model_gossipcop, X_train_gossipcop_resampled,␣
 ↪y_train_gossipcop_resampled)

print("\nSupport Vector Machine (Cross-Validation) (PolitiFact):")
svm_model_politifact = SVC(kernel='linear')
cross_validate_model(svm_model_politifact, X_train_politifact_resampled,␣
 ↪y_train_politifact_resampled)

# Naive Bayes
print("\nNaive Bayes (Cross-Validation) (GossipCop):")
nb_model_gossipcop = MultinomialNB()
cross_validate_model(nb_model_gossipcop, X_train_gossipcop_resampled,␣
 ↪y_train_gossipcop_resampled)

print("\nNaive Bayes (Cross-Validation) (PolitiFact):")
nb_model_politifact = MultinomialNB()
cross_validate_model(nb_model_politifact, X_train_politifact_resampled,␣
 ↪y_train_politifact_resampled)
```

```
Logistic Regression (Cross-Validation) (GossipCop):
Cross-Validation Scores: [0.78099694 0.8046788  0.83555041 0.82404748
0.82903981]
```

```
Average Cross-Validation Score: 0.8149

Logistic Regression (Cross-Validation) (PolitiFact):
Cross-Validation Scores: [0.84153005 0.8172043  0.84848485 0.83243243
0.85714286]
Average Cross-Validation Score: 0.8394

Support Vector Machine (Cross-Validation) (GossipCop):
Cross-Validation Scores: [0.77716995 0.80158282 0.83418669 0.82606989
0.82816229]
Average Cross-Validation Score: 0.8134

Support Vector Machine (Cross-Validation) (PolitiFact):
Cross-Validation Scores: [0.84324324 0.77419355 0.84102564 0.79569892
0.87292818]
Average Cross-Validation Score: 0.8254

Naive Bayes (Cross-Validation) (GossipCop):
Cross-Validation Scores: [0.77412321 0.79001628 0.79508493 0.77641007
0.77808832]
Average Cross-Validation Score: 0.7827

Naive Bayes (Cross-Validation) (PolitiFact):
Cross-Validation Scores: [0.85263158 0.83673469 0.86734694 0.84656085
0.87150838]
Average Cross-Validation Score: 0.8550
```

The results are consistent across the folds, which is a good sign that the models are generalizing well and not just memorizing the training data. The average performance metrics (F1-score) are very close to the fold-specific metrics. This consistency suggests that the models performed similarly on unseen data, that is another good sign.

1.9 Hyperparameter Tuning

My use of grid search and hyperparameter tuning was inspired by the methodology described by Chong et al. [17].

1.9.1 Hyperparameter Tuning Function

[11]:
```
# Hyperparameter tuning using Grid search - computationally intense, will run␣
↪for 1 h
# Define Parameter Grids
logistic_params = {
    'C': [0.1, 1, 10, 100],
    'solver': ['liblinear', 'saga']
}

svm_params = {
```

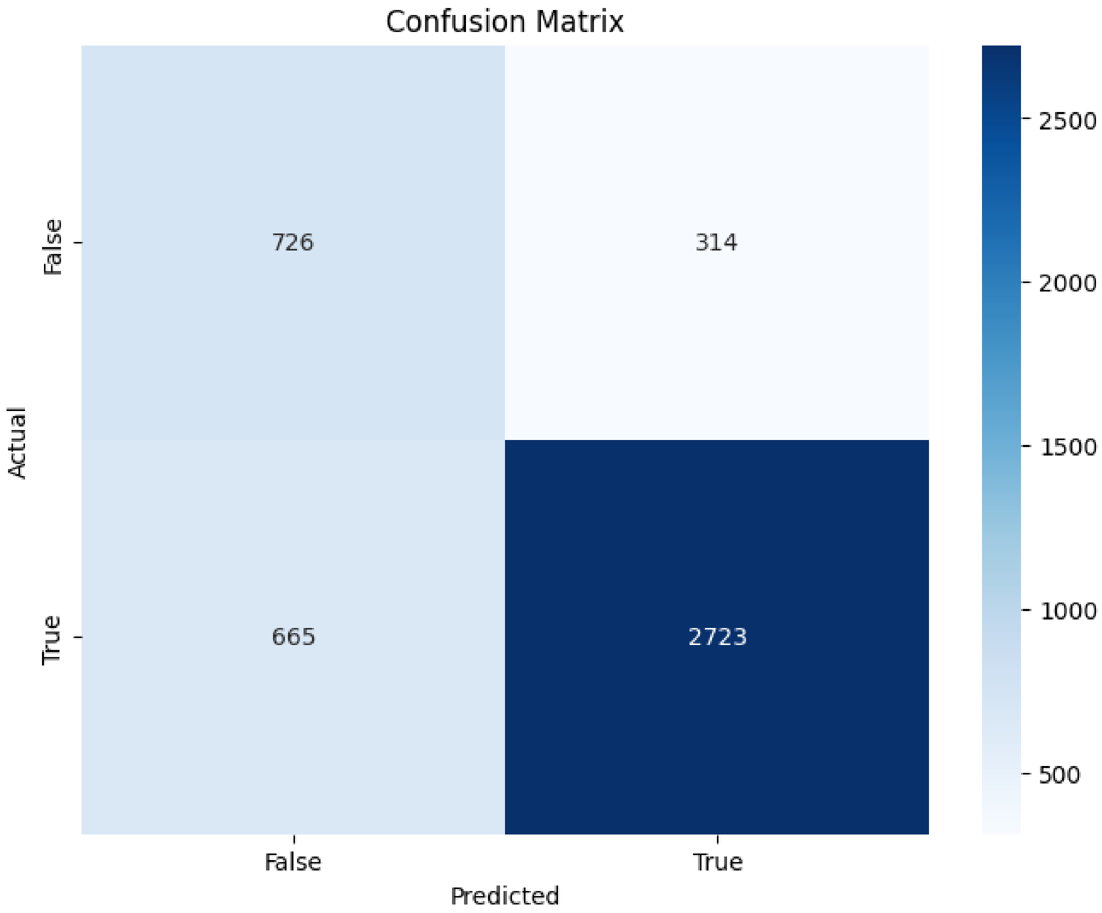
```
    'C': [0.1, 1, 10, 100],
    'kernel': ['linear', 'rbf']
}

nb_params = {
    'alpha': [0.1, 0.5, 1, 5, 10]
}

# Function to perform hyperparameter tuning and evaluation with 5-fold␣
↪cross-validation
def tune_and_evaluate(model, param_grid, X_train, X_test, y_train, y_test,␣
↪scoring='f1'):
    grid_search = GridSearchCV(model, param_grid, cv=5, scoring=scoring)
    grid_search.fit(X_train, y_train)
    best_model = grid_search.best_estimator_
    print(f'Best Parameters: {grid_search.best_params_}')
    # Evaluate the best model using cross-validation scores
    cv_results = grid_search.cv_results_
    mean_cv_score = grid_search.best_score_
    print(f'Best cross-validation score: {mean_cv_score:.4f}')
    # Evaluate the best model on the test set
    return train_and_evaluate(best_model, X_train, X_test, y_train, y_test)
```

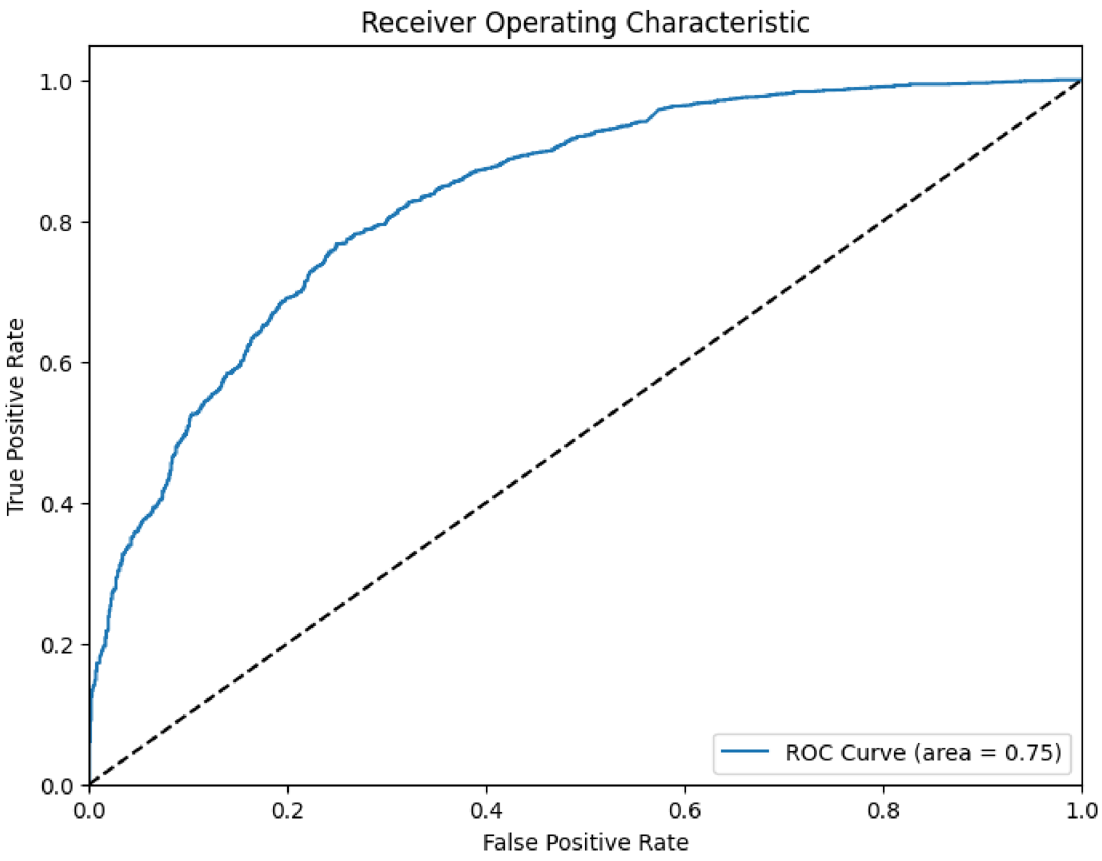
1.9.2 Hyperparameter Tuning Results

[12]:
```
# Logistic Regression with Hyperparameter Tuning
print("Logistic Regression with Hyperparameter Tuning (GossipCop):")
tune_and_evaluate(LogisticRegression(max_iter=1000, random_state=42),␣
↪logistic_params,
                  X_train_gossipcop_resampled, X_test_gossipcop_tfidf,␣
↪y_train_gossipcop_resampled, y_test_gossipcop)

print("\nLogistic Regression with Hyperparameter Tuning (PolitiFact):")
tune_and_evaluate(LogisticRegression(max_iter=1000, random_state=42),␣
↪logistic_params,
                  X_train_politifact_resampled, X_test_politifact_tfidf,␣
↪y_train_politifact_resampled, y_test_politifact)

# Support Vector Machine (SVM) with Hyperparameter Tuning
print("\nSupport Vector Machine with Hyperparameter Tuning (GossipCop):")
tune_and_evaluate(SVC(random_state=42), svm_params,
                  X_train_gossipcop_resampled, X_test_gossipcop_tfidf,␣
↪y_train_gossipcop_resampled, y_test_gossipcop)

print("\nSupport Vector Machine with Hyperparameter Tuning (PolitiFact):")
tune_and_evaluate(SVC(random_state=42), svm_params,
```

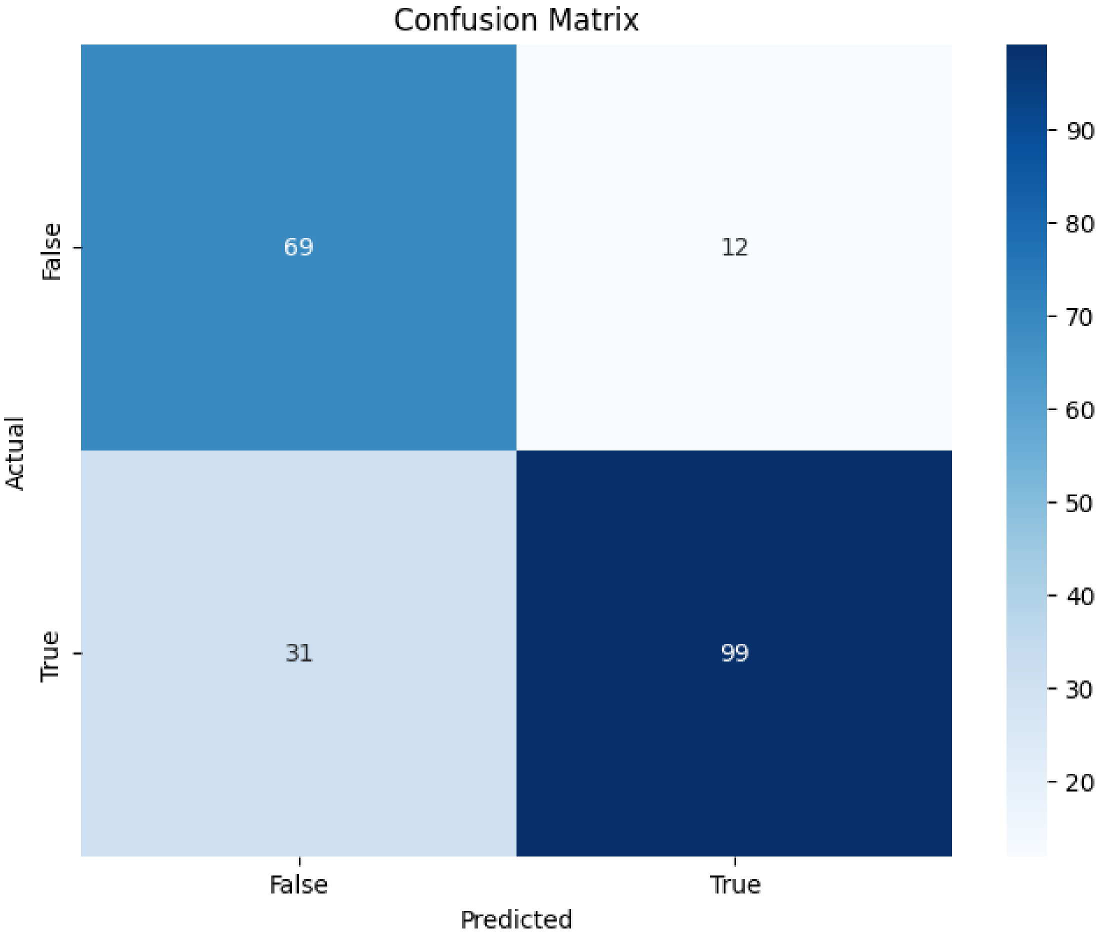
```
                  X_train_politifact_resampled, X_test_politifact_tfidf,␣
 ↪y_train_politifact_resampled, y_test_politifact)

# Naive Bayes with Hyperparameter Tuning
print("\nNaive Bayes with Hyperparameter Tuning (GossipCop):")
tune_and_evaluate(MultinomialNB(), nb_params,
                  X_train_gossipcop_resampled, X_test_gossipcop_tfidf,␣
 ↪y_train_gossipcop_resampled, y_test_gossipcop)

print("\nNaive Bayes with Hyperparameter Tuning (PolitiFact):")
tune_and_evaluate(MultinomialNB(), nb_params,
                  X_train_politifact_resampled, X_test_politifact_tfidf,␣
 ↪y_train_politifact_resampled, y_test_politifact)
```

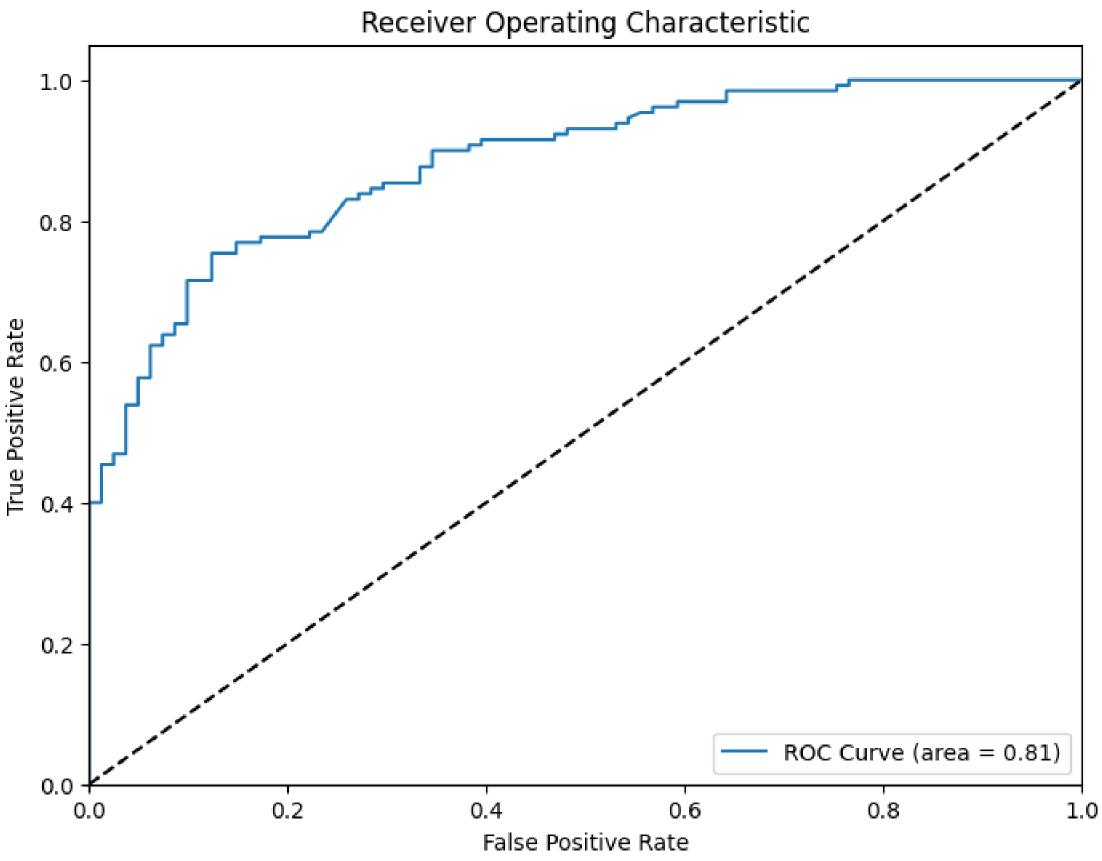
```
Logistic Regression with Hyperparameter Tuning (GossipCop):
Best Parameters: {'C': 10, 'solver': 'liblinear'}
Best cross-validation score: 0.8158
Accuracy: 0.7789
Precision: 0.8966
Recall: 0.8037
F1-score: 0.8476
ROC AUC: 0.7509
Confusion Matrix:
[[ 726  314]
 [ 665 2723]]
```


Confusion Matrix
726
314
665
2723
False
True
Actual
False
True
Predicted
2500
2000
1500
1000
500

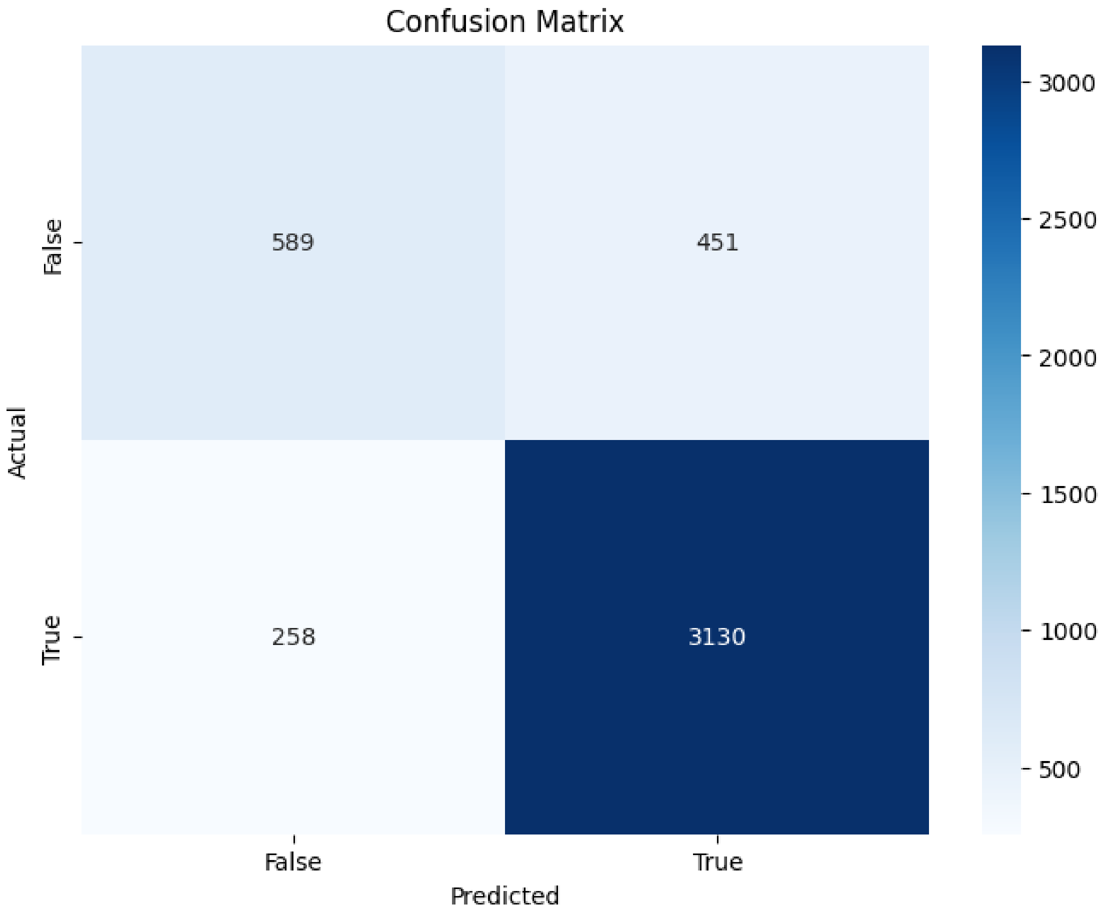


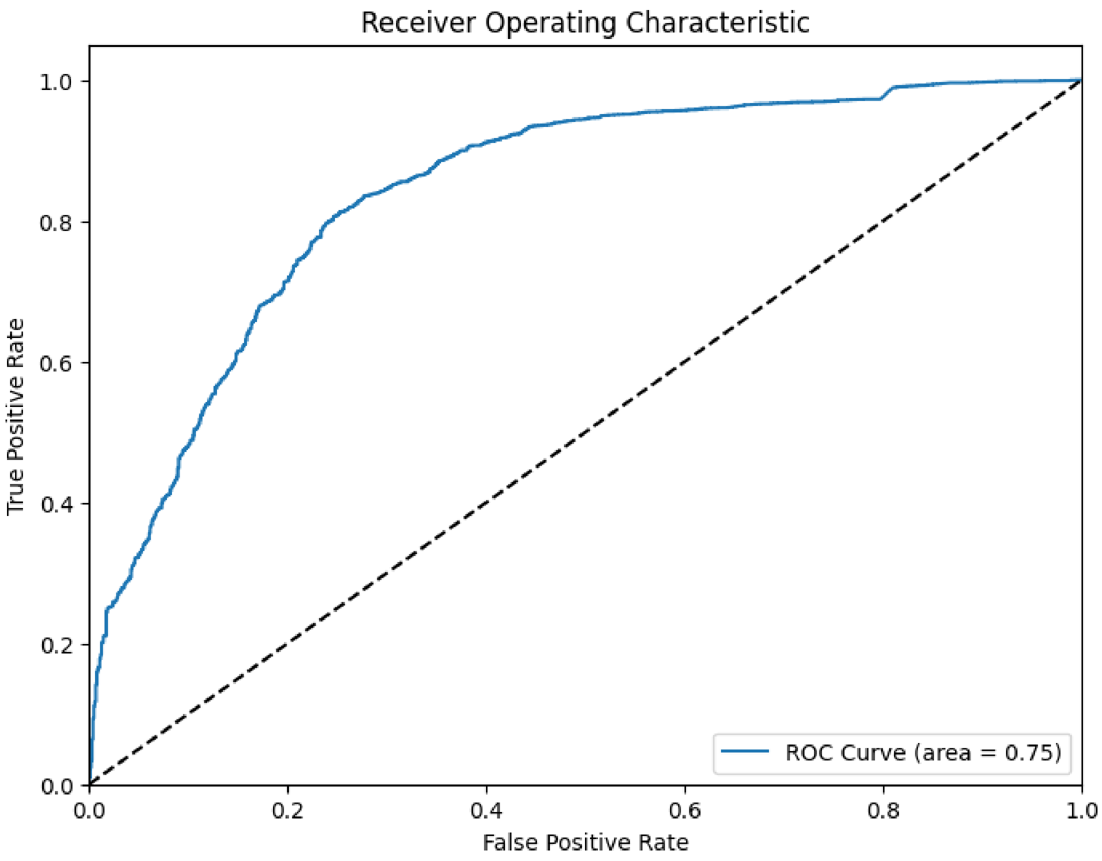
```
Logistic Regression with Hyperparameter Tuning (PolitiFact):
Best Parameters: {'C': 1, 'solver': 'liblinear'}
Best cross-validation score: 0.8394
Accuracy: 0.7962
Precision: 0.8919
Recall: 0.7615
F1-score: 0.8216
ROC AUC: 0.8067
Confusion Matrix:
[[69 12]
 [31 99]]
```


Confusion Matrix
69
12
31
99
False
True
Actual
False
True
Predicted
90
80
70
60
50
40
30
20

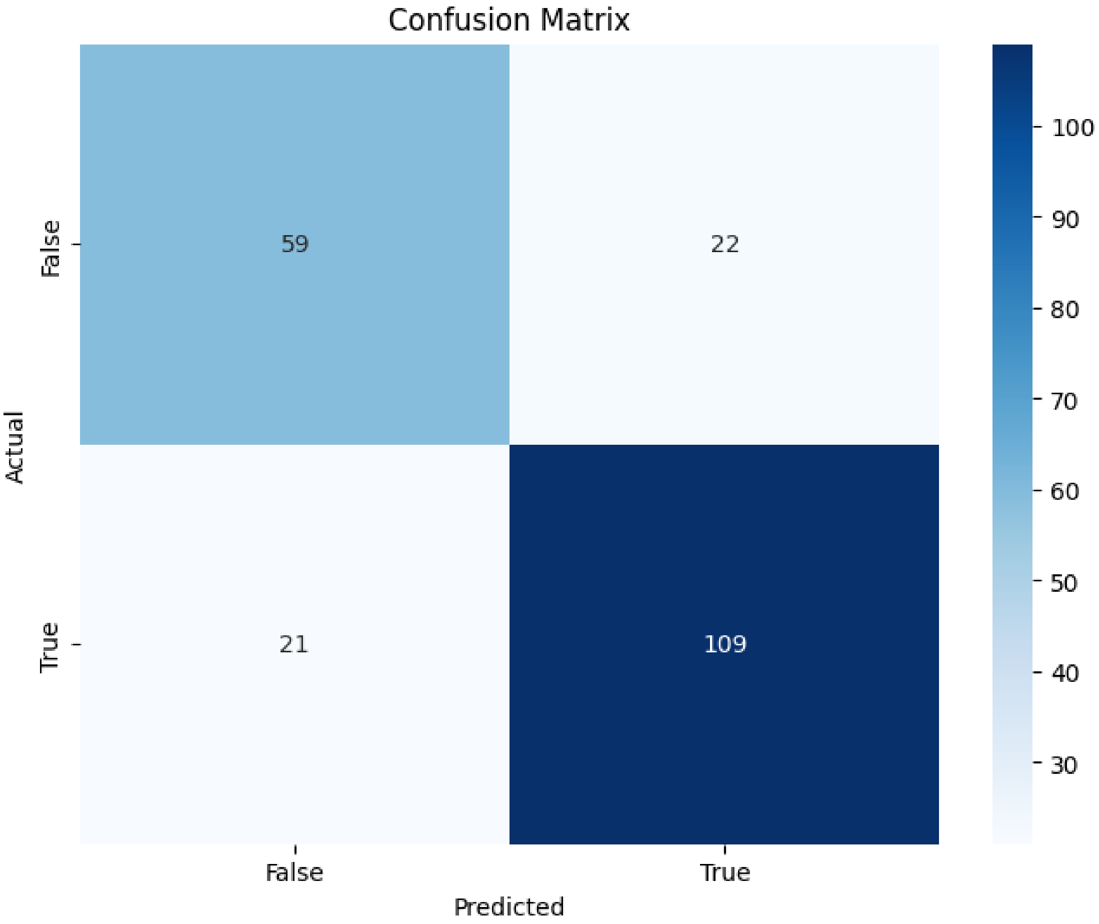


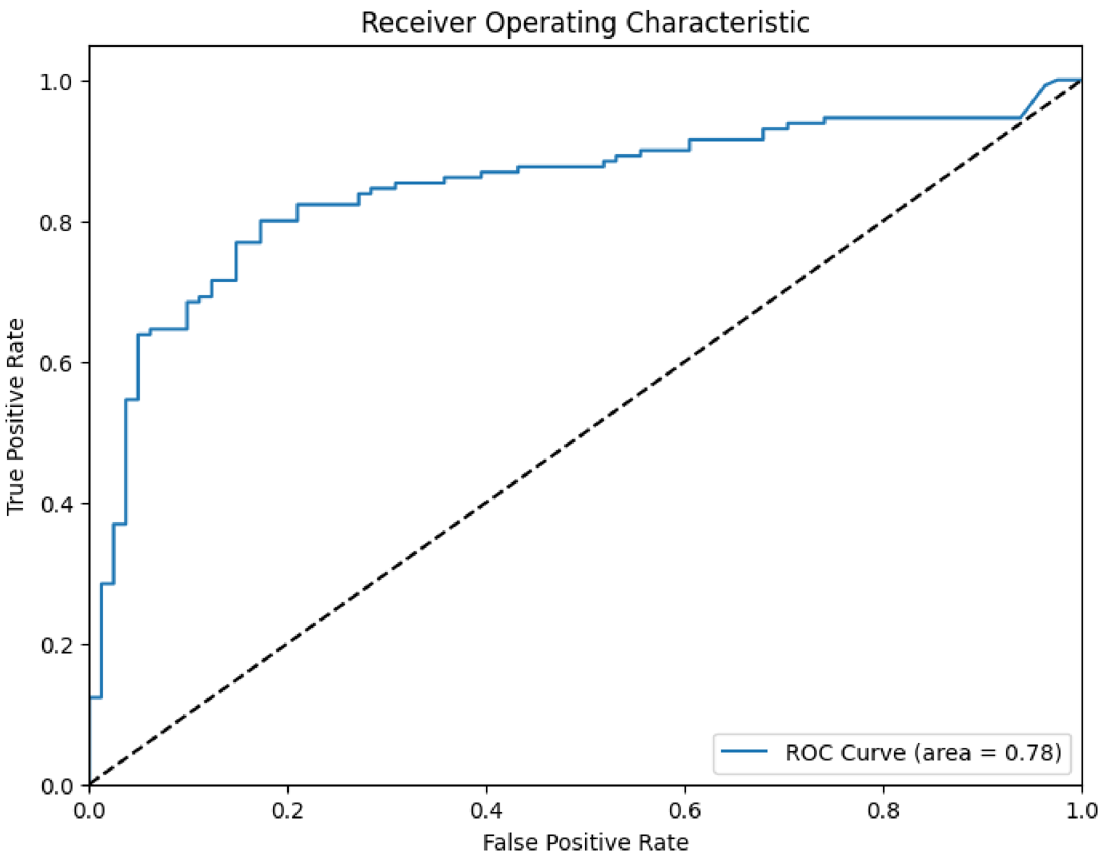
```
Support Vector Machine with Hyperparameter Tuning (GossipCop):
Best Parameters: {'C': 10, 'kernel': 'rbf'}
Best cross-validation score: 0.9207
Accuracy: 0.8399
Precision: 0.8741
Recall: 0.9238
F1-score: 0.8983
ROC AUC: 0.7451
Confusion Matrix:
[[ 589  451]
 [ 258 3130]]
```


Confusion Matrix
Actual
False
True
Predicted
False
True
589
451
258
3130
3000
2500
2000
1500
1000
500

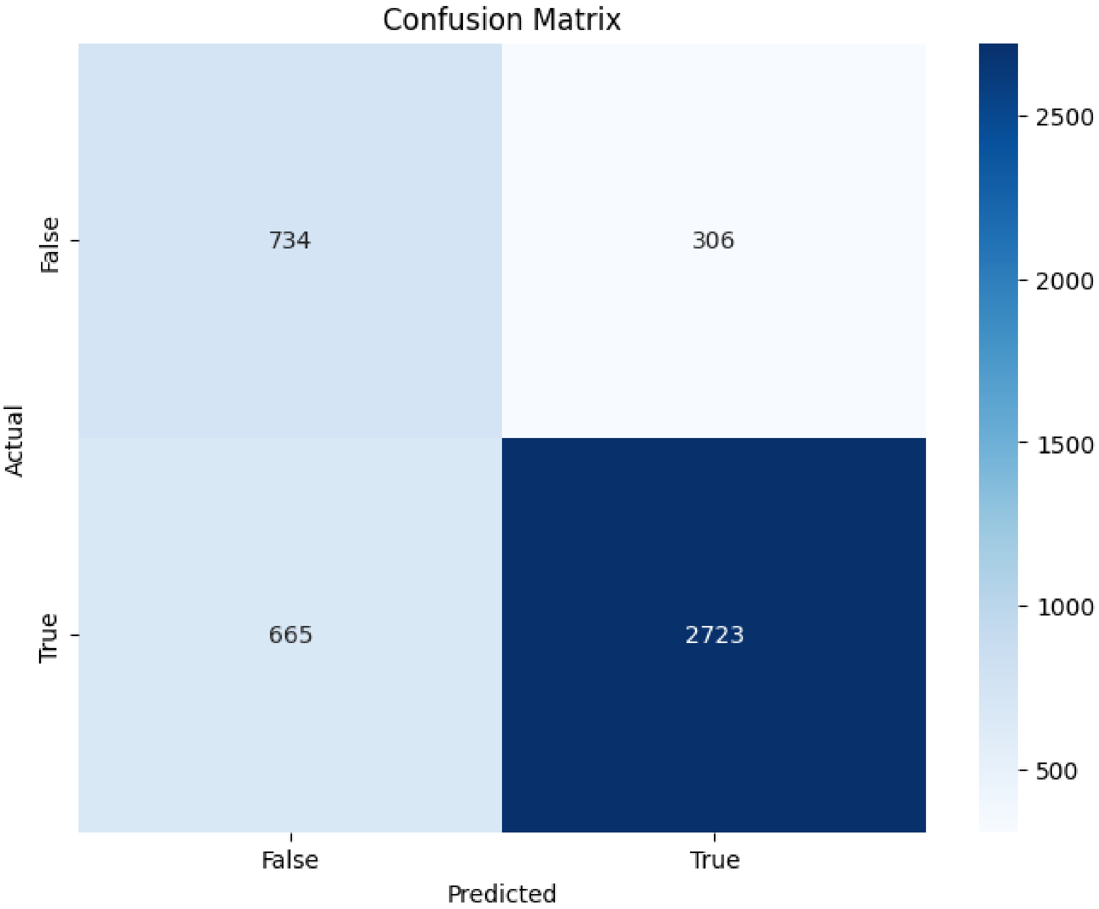


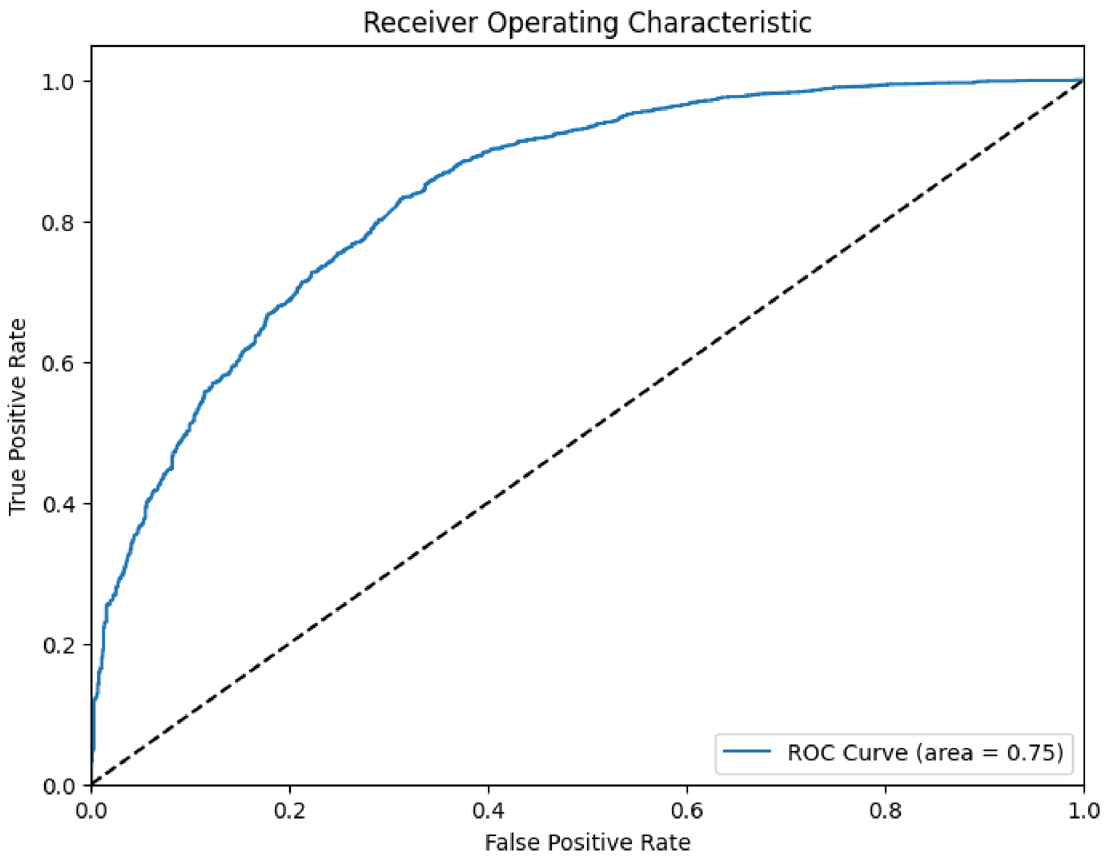
```
Support Vector Machine with Hyperparameter Tuning (PolitiFact):
Best Parameters: {'C': 1, 'kernel': 'rbf'}
Best cross-validation score: 0.8650
Accuracy: 0.7962
Precision: 0.8321
Recall: 0.8385
F1-score: 0.8352
ROC AUC: 0.7834
Confusion Matrix:
[[ 59  22]
 [ 21 109]]
```


Confusion Matrix
59
22
21
109
False
True
Actual
False
True
Predicted
100
90
80
70
60
50
40
30

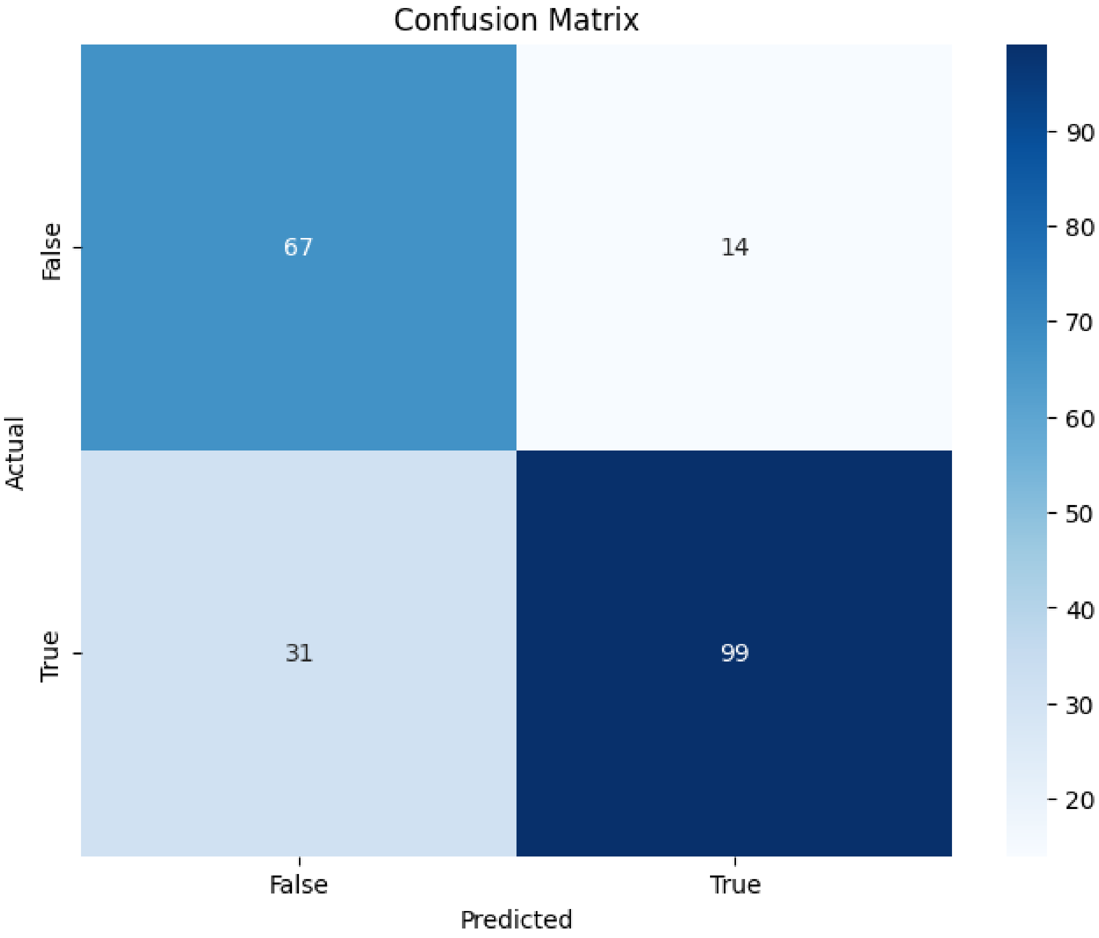


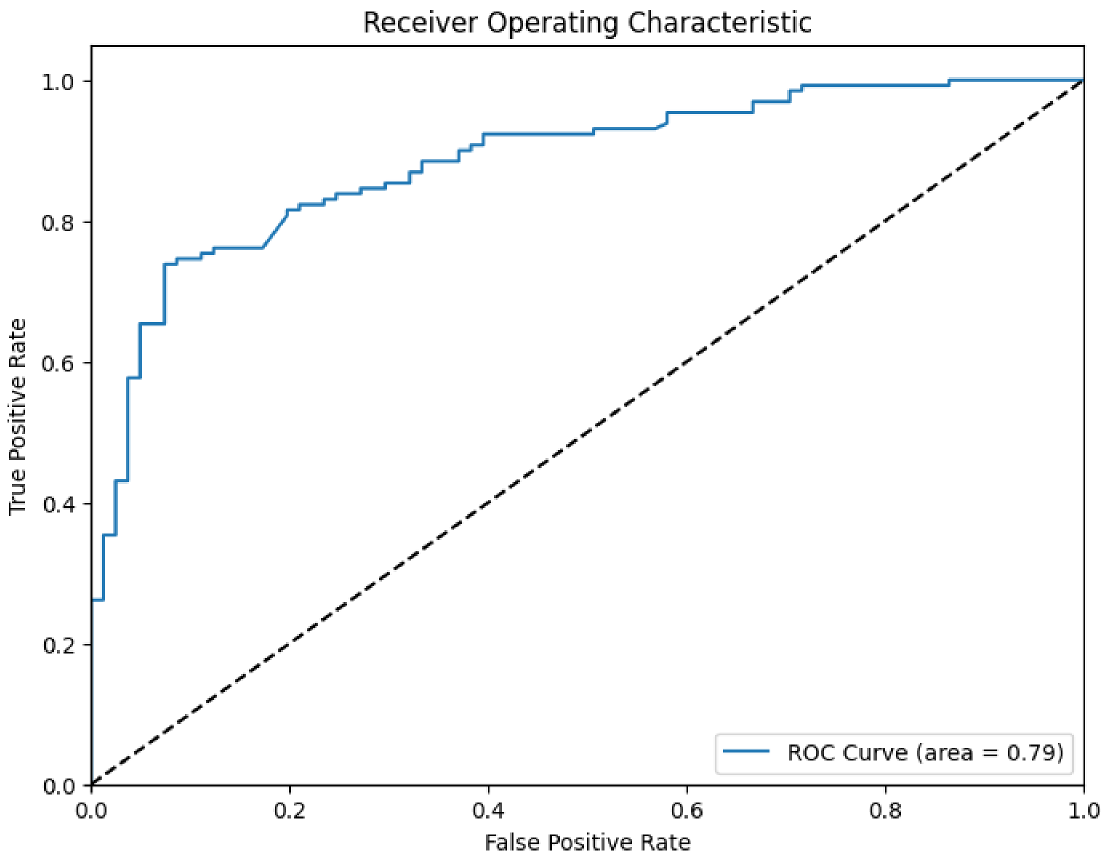
```
Naive Bayes with Hyperparameter Tuning (GossipCop):
Best Parameters: {'alpha': 0.1}
Best cross-validation score: 0.7837
Accuracy: 0.7807
Precision: 0.8990
Recall: 0.8037
F1-score: 0.8487
ROC AUC: 0.7547
Confusion Matrix:
[[ 734  306]
 [ 665 2723]]
```


Confusion Matrix
734
306
665
2723
False
True
Actual
False
True
Predicted
2500
2000
1500
1000
500



```
Naive Bayes with Hyperparameter Tuning (PolitiFact):
Best Parameters: {'alpha': 1}
Best cross-validation score: 0.8550
Accuracy: 0.7867
Precision: 0.8761
Recall: 0.7615
F1-score: 0.8148
ROC AUC: 0.7943
Confusion Matrix:
[[67 14]
 [31 99]]
```


Confusion Matrix
67
14
31
99
False
True
Actual
False
True
Predicted
90
80
70
60
50
40
30
20



```
[12]: (MultinomialNB(alpha=1),
        array([1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1,
               0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0,
               0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
               1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0,
               1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
               1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1,
               0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0,
               1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1,
               0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
               0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]))
```

Logistic Regression

GossipCop:

```
Original:
    Accuracy: 0.7850
    F1-score: 0.8520
    ROC AUC: 0.7579
Fine-Tuned:
```

```
    Accuracy: 0.7789 (slightly worse)
    F1-score: 0.8476 (slightly worse)
    ROC AUC: 0.7509 (worse)
```

PolitiFact:

```
Both original and fine-tuned models have identical metrics.
```

Support Vector Machine

GossipCop:

```
Original:
    Accuracy: 0.7764
    F1-score: 0.8449
    ROC AUC: 0.7543
Fine-Tuned:
    Accuracy: 0.8399 (improved)
    F1-score: 0.8983 (improved)
    ROC AUC: 0.7451 (slightly worse)
```

PolitiFact:

```
Original:
    Accuracy: 0.7820
    F1-score: 0.8099
    ROC AUC: 0.7905
Fine-Tuned:
    Accuracy: 0.7962 (improved)
    F1-score: 0.8352 (improved)
    ROC AUC: 0.7834 (slightly worse)
```

Naive Bayes

GossipCop:

```
Original:
    Accuracy: 0.7809
    F1-score: 0.8485
    ROC AUC: 0.7572
Fine-Tuned:
    Accuracy: 0.7807 (about the same)
    F1-score: 0.8487 (about the same)
    ROC AUC: 0.7547 (slightly worse)
```

PolitiFact:

```
Both original and fine-tuned models have identical metrics.
```

Hyperparameter tuning has led to some improvements, especially in the case of SVM on the GossipCop dataset. However, for Logistic Regression and Naive Bayes, the changes are minimal.

1.10 Save the Best Model to Google Drive

[13]:
```
# Selecting svm_model_gossipcop as the best model after hyperparameter tuning
best_model = SVC(C=10, kernel='rbf', random_state=42)
best_model.fit(X_train_gossipcop_resampled, y_train_gossipcop_resampled)

# Save the model to a file
model_filename = '/content/drive/My Drive/best_svm_model_gossipcop.pkl'
joblib.dump(best_model, model_filename)

# Save the TF-IDF vectorizer to a file
vectorizer_filename = '/content/drive/My Drive/tfidf_vectorizer_gossipcop.pkl'
joblib.dump(vectorizer, vectorizer_filename)

print("Model and vectorizer saved to Google Drive.")
```

```
Model and vectorizer saved to Google Drive.
```

1.11 Conclusion

My models significantly outperform the baseline results across all metrics for both datasets (GossipCop and PolitiFact). Balancing the datasets with SMOTE and tuning hyperparameters further improved the models' performance.

Based on the evaluation metrics (accuracy, precision, recall, F1-score, and ROC AUC) for each classifier, the Support Vector Machine (SVM) with hyperparameter tuning on the GossipCop dataset seems to perform the best overall.

1.12 References

[5] BIAN, T., XIAO, X., XU, T., ZHAO, P., HUANG, W., RONG, Y., and HUANG, J. 2020. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. arXiv preprint arXiv:2001.06362. https://doi.org/10.48550/arXiv.2001.06362

[6] SHEN, Y., LIU, Q., GUO, N., YUAN, J., and YANG, Y. 2023. Fake News Detection on Social Networks: A Survey. Applied Sciences, 13, 11877. https://doi.org/10.3390/app132111877

[7] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. https://doi.org/10.48550/arXiv.1809.01286

[14] CHAWLA, N.V., BOWYER, K.W., HALL, L.O., and KEGELMEYER, W.P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.48550/arXiv.1106.1813

[15] BROWNE, M.W. 2000. Cross-Validation Methods. Journal of Mathematical Psychology, 44, 108–132. https://doi.org/10.1006/jmps.1999.1279

[16] POWERS, D.M.W. 2011. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. International Journal of Machine Learning Technology, 2(1), pp. 37-63. https://doi.org/10.48550/arXiv.2010.16061

[17] CHONG, K. and SHAH, N. 2022. Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis. International Journal of Advanced Computer Science and Applications, 13(12), pp. 90-94. DOI: 10.14569/IJACSA.2022.0131213

[ ]:

```
# Install LaTeX packages necessary for converting notebooks to PDF
!apt-get update
!apt-get install -y texlive-xetex texlive-fonts-recommended␣
 ↪texlive-plain-generic texlive-latex-extra pandoc

# Convert the notebook to PDF
!jupyter nbconvert --to pdf "/content/drive/My Drive/Colab Notebooks/
 ↪FakeNewsNetClassifier.ipynb"
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,626 B]
Hit:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:7 https://r2u.stat.illinois.edu/ubuntu jammy Release [5,713 B]
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Get:9 https://r2u.stat.illinois.edu/ubuntu jammy Release.gpg [793 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages
[1,998 kB]
Get:11 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease [24.3 kB]
Hit:12 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:13 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:14 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,181 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[1,410 kB]
Get:16 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy/main
amd64 Packages [48.1 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
[1,127 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,263
kB]
Get:19 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,544 kB]
Fetched 17.9 MB in 4s (4,259 kB/s)
Reading package lists… Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
```

```
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc-data
poppler-data
  preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick ruby-
xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base texlive-
binaries
  texlive-latex-base texlive-latex-recommended texlive-pictures tipa xfonts-
encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java texlive-
luatex
  pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc nodejs php python
libjs-mathjax
  libjs-katex citation-style-language-styles poppler-utils ghostscript fonts-
japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-
arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv | postscript-
viewer perl-tk xpdf
  | pdf-viewer xzdec texlive-fonts-recommended-doc texlive-latex-base-doc
python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-
extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex texlive-
pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc
pandoc-data
```

```
  poppler-data preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-
webrick ruby-xmlrpc
  ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
texlive-binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-extra texlive-
latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa xfonts-encodings
xfonts-utils
0 upgraded, 58 newly installed, 0 to remove and 45 not upgraded.
Need to get 202 MB of archives.
After this operation, 728 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.7 [752 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.7 [5,028 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-
gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-gfm-
```

```
extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.7 [50.1 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
Get:28 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ruby-webrick all
1.7.0-3 [51.8 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.7 [5,113 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc-data all
2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc amd64
2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
```

```
all 12.2-1ubuntu1 [185 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 17s (11.9 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 123576 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
```

```
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.7_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.7) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.7_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.7) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack …/17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack …/18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
```

```
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/19-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/20-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/21-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/23-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/24-ruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/25-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/26-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/27-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/28-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/29-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/31-libruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/32-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/33-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
```

```
Preparing to unpack …/34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/36-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/38-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/39-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package pandoc-data.
Preparing to unpack …/40-pandoc-data_2.9.2.1-3ubuntu2_all.deb …
Unpacking pandoc-data (2.9.2.1-3ubuntu2) …
Selecting previously unselected package pandoc.
Preparing to unpack …/41-pandoc_2.9.2.1-3ubuntu2_amd64.deb …
Unpacking pandoc (2.9.2.1-3ubuntu2) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/42-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/43-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/44-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/45-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/47-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/48-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
```

```
Preparing to unpack …/49-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/50-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/51-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/52-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/53-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/54-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/55-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/56-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/57-texlive-xetex_2021.20220204-1_all.deb …
Unpacking texlive-xetex (2021.20220204-1) …
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
Setting up tex-common (6.17) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
```

```
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up ruby-webrick (1.7.0-3) …
Setting up libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up pandoc-data (2.9.2.1-3ubuntu2) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.7) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.7) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up pandoc (2.9.2.1-3ubuntu2) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
```

```
Setting up tipa (2:1.3-21) …
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Setting up ruby3.0 (3.0.2-7ubuntu2.7) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.4) …
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

Processing triggers for tex-common (6.17) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time…
```

FakeNewsNetCNN

September 8, 2024

1 FakeNewsNetCNN

This notebook builds and evaluates Convolutional Neural Network (CNN) model for detecting fake news using the FakeNewsNet dataset, that contians two subsets, GossipCop and PolitiFact. This work is based on the work of Shu et al. [7] and Denny Britz repository. [11]

1.1 Import necessary libraries

[4]:

```
import numpy as np
import random
import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score, roc_auc_score, confusion_matrix, classification_report, roc_curve
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, KFold
from sklearn.utils import class_weight
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv1D, MaxPooling1D,␣
 ↪GlobalMaxPooling1D, Flatten, Dropout, Embedding
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
from tensorflow.keras.backend import clear_session

# Check and install keras-tuner if not available
try:
    import keras_tuner as kt
except ImportError:
    !pip install keras-tuner
    import keras_tuner as kt
```

1.2 Add Reproducibility logic

Functions to set seeds and preserve deterministic operations are defined for reproducibility. Different runs have the same results.

[5]:

```python
# Set seeds for reproducibility
def set_seeds(seed=42):
    np.random.seed(seed)
    random.seed(seed)
    tf.random.set_seed(seed)

set_seeds()

# Ensure deterministic operations
def set_deterministic():
    tf.config.threading.set_intra_op_parallelism_threads(1)
    tf.config.threading.set_inter_op_parallelism_threads(1)

# Call set_deterministic before any TensorFlow operations
set_deterministic()
```

1.3 Load Preprocessed Data from Google Drive

[6]:

```python
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Load the preprocessed datasets from Google Drive
gossipcop_combined = pd.read_csv('/content/drive/My Drive/
 ↪gossipcop_preprocessed.csv')
politifact_combined = pd.read_csv('/content/drive/My Drive/
 ↪politifact_preprocessed.csv')

# Quick check of the data loaded
print(gossipcop_combined.head())
print(politifact_combined.head())
```

```
Mounted at /content/drive
                                               title  label
0  lea michel hairstylist mix textur spray coconu…      1
1  thoma markl princ harri polit miss daughter me…      0
2  2019 sag award nomin see full list nomine varieti      1
3  see meghan markl royal coat arm symbol hide wi…      1
4    kyli jenner visit shaman life kyli season final      1
                                               title  label
0         world popular candi remov shelv octob 2017      0
1                             brows congression bill      1
2                          suprem court vacanc video      1
```

```
3                                    u import export      1
4  die 78 year old cia agent admit kill marilyn m…      0
```

1.4 Check loaded data

[7]:

```
# Check for NaN values in datasets
print("NaN values in GossipCop dataset:\n", gossipcop_combined.isnull().sum())
print("NaN values in PolitiFact dataset:\n", politifact_combined.isnull().sum())

# Ensure there are no NaN values
gossipcop_combined.dropna(inplace=True)
politifact_combined.dropna(inplace=True)

# Re-check for NaN values in datasets
print("NaN values in GossipCop dataset after dropna:\n", gossipcop_combined.
 ↪isnull().sum())
print("NaN values in PolitiFact dataset after dropna:\n", politifact_combined.
 ↪isnull().sum())
```

```
NaN values in GossipCop dataset:
 title    1
label    0
dtype: int64
NaN values in PolitiFact dataset:
 title    2
label    0
dtype: int64
NaN values in GossipCop dataset after dropna:
 title    0
label    0
dtype: int64
NaN values in PolitiFact dataset after dropna:
 title    0
label    0
dtype: int64
```

1.5 Prepare Data for CNN

Features (X) and labels (y) are defined for both datasets. Text data is tokenized, padded, and split into training and testing sets. Tokenization converts text into numerical values. This process makes it suitable for neural network operations. Padding keeps uniform input length and enables efficient batch processing as explained by Denny Britz blog post linked to his repository[11]. Encoding labels into numerical form standardizes the output for classification tasks. I am splitting the dataset the same way as Shu et al. [7]: “We use 80% of data for training and 20% for testing.”

[8]:

```
# Define features (X) and labels (y)
X_gossipcop = gossipcop_combined['title'].values
y_gossipcop = gossipcop_combined['label'].values
```

```
X_politifact = politifact_combined['title'].values
y_politifact = politifact_combined['label'].values

# Function to tokenize and pad sequences
def tokenize_and_pad(texts, max_num_words=5000, maxlen=100):
    tokenizer = Tokenizer(num_words=max_num_words)
    tokenizer.fit_on_texts(texts)
    sequences = tokenizer.texts_to_sequences(texts)
    padded_sequences = pad_sequences(sequences, padding='post', maxlen=maxlen)
    return padded_sequences, tokenizer

# Tokenize and pad the sequences for both datasets
X_gossipcop, tokenizer_gossipcop = tokenize_and_pad(X_gossipcop)
X_politifact, tokenizer_politifact = tokenize_and_pad(X_politifact)

# Encode the labels as integers
def encode_labels(labels):
    le = LabelEncoder()
    encoded_labels = le.fit_transform(labels)
    return encoded_labels, le

y_gossipcop, le_gossipcop = encode_labels(y_gossipcop)
y_politifact, le_politifact = encode_labels(y_politifact)

# Split the data into training and testing sets for both datasets with a test␣
 ↪size of 20%
def split_data(X, y, test_size=0.2, random_state=42):
    return train_test_split(X, y, test_size=test_size,␣
 ↪random_state=random_state)

X_train_gossipcop, X_test_gossipcop, y_train_gossipcop, y_test_gossipcop =␣
 ↪split_data(X_gossipcop, y_gossipcop)
X_train_politifact, X_test_politifact, y_train_politifact, y_test_politifact =␣
 ↪split_data(X_politifact, y_politifact)
```

1.6 Build the CNN Model

Separate CNN models for GossipCop and PolitiFact are built using embedding, convolutional, pooling, dropout, and dense layers. Models are compiled with Adam optimizer and binary crossentropy loss. I have experimented with different layers, parameters and settings explained by Denny Britz [11].

[9]:

```
# Build the CNN Model for GossipCop
def create_cnn_model_gossipcop(input_length):
    model = Sequential([
        Embedding(input_dim=5000, output_dim=128, input_length=input_length), ␣
 ↪# Embedding layer
```

```
        Conv1D(filters=64, kernel_size=5, activation='relu', kernel_regularizer=l2(0.01)),  # Conv1D layer
        MaxPooling1D(pool_size=2),  # MaxPooling layer
        Conv1D(filters=32, kernel_size=5, activation='relu', kernel_regularizer=l2(0.01)),  # Conv1D layer
        GlobalMaxPooling1D(),  # GlobalMaxPooling layer
        Dense(units=32, activation='relu'),  # Dense layer
        Dropout(0.5),  # Dropout layer
        Dense(units=1, activation='sigmoid')  # Output layer
    ])
    model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])
    return model

# Build the CNN Model for PolitiFact
def create_cnn_model_politifact(input_length):
    model = Sequential([
        Embedding(input_dim=5000, output_dim=128, input_length=input_length),  # Embedding layer
        Conv1D(filters=128, kernel_size=5, activation='relu', kernel_regularizer=l2(0.01)),  # Conv1D layer
        MaxPooling1D(pool_size=2),  # MaxPooling layer
        Conv1D(filters=64, kernel_size=5, activation='relu', kernel_regularizer=l2(0.01)),  # Conv1D layer
        GlobalMaxPooling1D(),  # GlobalMaxPooling layer
        Dense(units=64, activation='relu', kernel_regularizer=l2(0.01)),  # Dense layer
        Dropout(0.6),  # Dropout layer
        Dense(units=1, activation='sigmoid', kernel_regularizer=l2(0.01))  # Output layer
    ])
    model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])
    return model

input_length = X_train_gossipcop.shape[1]

cnn_model_gossipcop = create_cnn_model_gossipcop(input_length)
cnn_model_politifact = create_cnn_model_politifact(input_length)

# Callbacks to prevent overfitting and make training more efficient
early_stopping_gossipcop = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True) # Stop training when a monitored metric has stopped improving
early_stopping_politifact = EarlyStopping(monitor='val_loss', patience=7, restore_best_weights=True)
```

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=3,␣
 ↪min_lr=0.00001) # Reduce the learning rate when a metric has stopped␣
 ↪improving
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

1.7 Handle Class Imbalance

Class weights are calculated to handle class imbalance in the datasets. These weights are used during model training to balance the impact of each class.

[10]:

```
# Function to compute class weights and return as a dictionary
def compute_class_weights(y_train):
    class_weights = class_weight.compute_class_weight(class_weight='balanced',␣
 ↪classes=np.unique(y_train), y=y_train)
    class_weights_dict = {i: class_weights[i] for i in␣
 ↪range(len(class_weights))}
    return class_weights_dict

# Calculate class weights for GossipCop
class_weights_dict_gossipcop = compute_class_weights(y_train_gossipcop)

# Calculate class weights for PolitiFact
class_weights_dict_politifact = compute_class_weights(y_train_politifact)
```

1.8 Train and Evaluate CNN Model

The CNN models are trained with early stopping and learning rate reduction callbacks. Performance metrics accuracy, precision, recall, F1-score, and ROC AUC are calculated and displayed, also with confusion matrices and ROC curves.

According to Powers et al.[16], "The F1-score, which is the harmonic mean of precision and recall, is particularly recommended for imbalanced datasets because it provides a balance between false positives and false negatives, thus giving a more comprehensive measure of a model's performance." so I will use F1 score during training and evaluation.

Keras doesn't natively support the F1 score as a metric during training. I will keep accuracy as a metric for monitoring during training because it's fast and gives a general sense of model performance. I will use a custom callback to log the F1 score at the end of each epoch.

[11]:

```
# Custom callback to log the F1 score
class F1ScoreCallback(tf.keras.callbacks.Callback):
    def __init__(self, validation_data):
        self.validation_data = validation_data
        self.f1_scores = []
        self.precision_scores = []
        self.recall_scores = []
```

```
    def on_epoch_end(self, epoch, logs=None):
        val_pred = (self.model.predict(self.validation_data[0]) > 0.5).
 ↪astype("int32")
        val_true = self.validation_data[1]
        val_f1 = f1_score(val_true, val_pred)
        val_precision = precision_score(val_true, val_pred)
        val_recall = recall_score(val_true, val_pred)
        self.f1_scores.append(val_f1)
        self.precision_scores.append(val_precision)
        self.recall_scores.append(val_recall)
        print(f' - val_f1: {val_f1:.4f} - val_precision: {val_precision:.4f} -␣
 ↪val_recall: {val_recall:.4f}')
```

[12]:

```
# Train and evaluate the model, returning the model, predictions and history
def train_and_evaluate(model, X_train, X_test, y_train, y_test, class_weights,␣
 ↪early_stopping, reduce_lr, train_model=True):
    f1_callback = F1ScoreCallback(validation_data=(X_test, y_test))
    history = None

    if train_model:
        # Train the model
        history = model.fit(X_train, y_train, epochs=50, batch_size=64,
                            validation_split=0.2, class_weight=class_weights,␣
 ↪verbose=1,
                            callbacks=[early_stopping, reduce_lr, f1_callback])
        # Update history to include F1, precision, and recall
        history.history['val_f1_score'] = f1_callback.f1_scores
        history.history['val_precision'] = f1_callback.precision_scores
        history.history['val_recall'] = f1_callback.recall_scores

    # Make predictions
    y_pred_prob = model.predict(X_test)
    y_pred = (y_pred_prob > 0.5).astype("int32")

    # Calculate metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    roc_auc = roc_auc_score(y_test, y_pred_prob)

    # Print and plot evaluation metrics and figures
    print(f'Accuracy: {accuracy:.4f}')
    print(f'Precision: {precision:.4f}')
    print(f'Recall: {recall:.4f}')
    print(f'F1-score: {f1:.4f}')
```

```
    print(f'ROC AUC: {roc_auc:.4f}')

    cm = confusion_matrix(y_test, y_pred)
    print('Confusion Matrix:')
    print(cm)

    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['False',␣
  ↪'True'], yticklabels=['False', 'True'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()

    fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
    plt.figure(figsize=(8, 6))
    plt.plot(fpr, tpr, label=f'ROC Curve (area = {roc_auc:.2f})')
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC Curve')
    plt.legend(loc="lower right")
    plt.show()

    return model, y_pred, history
```

1.8.1 Model Training and Evaluation

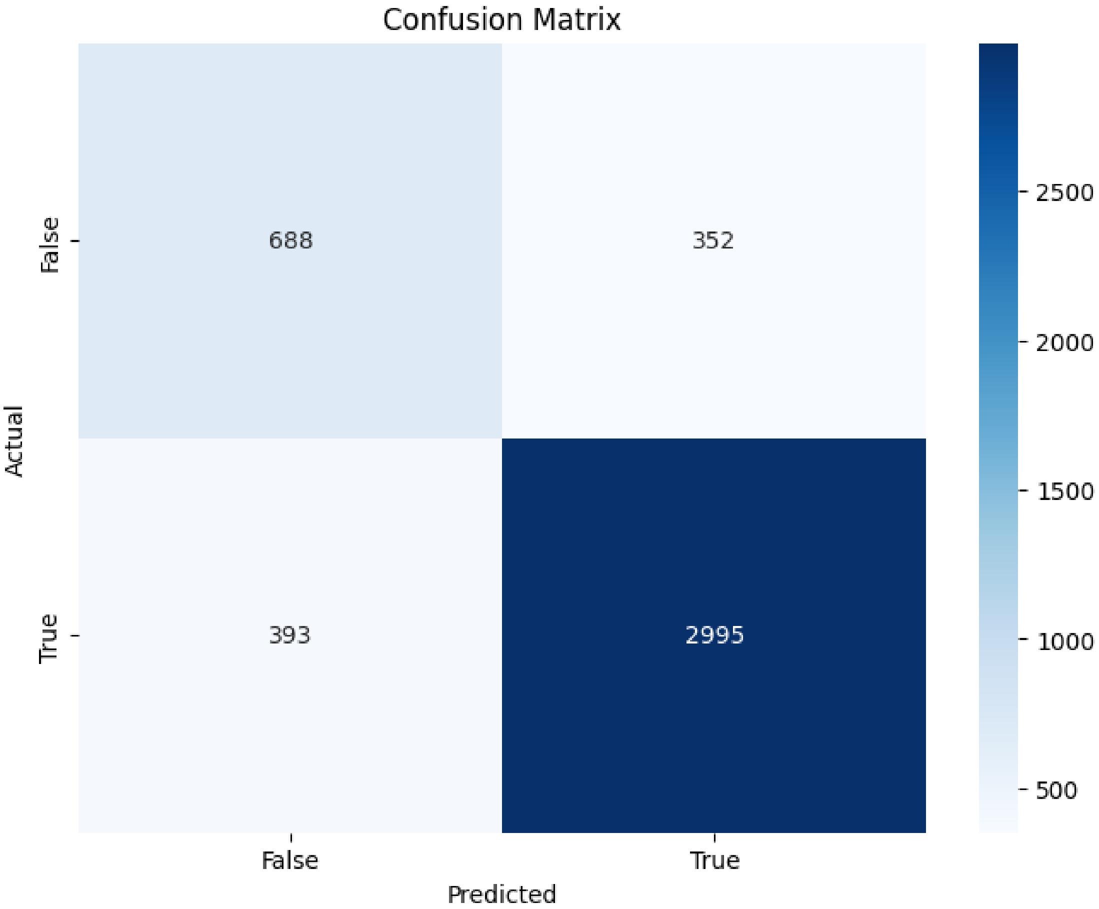
[13]:

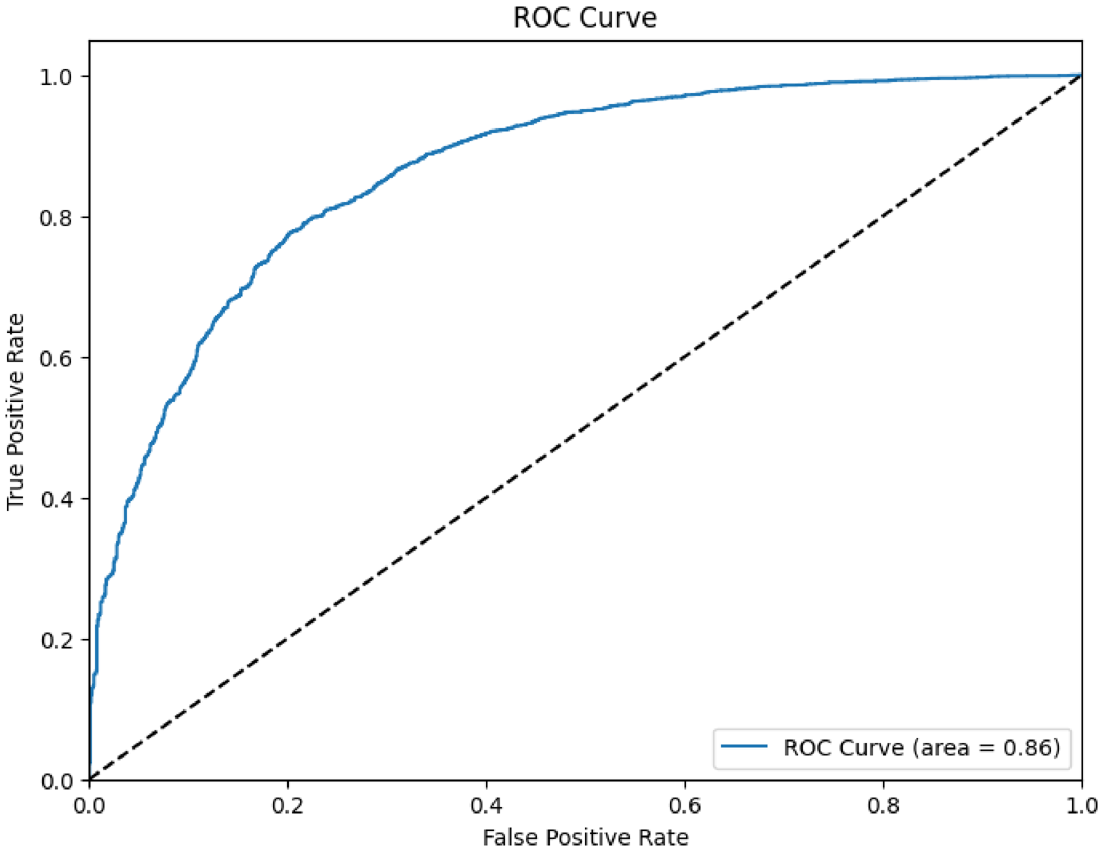
```
# Train and evaluate the CNN model for GossipCop dataset
cnn_model_gossipcop, y_pred_gossipcop, history_gossipcop =␣
  ↪train_and_evaluate(cnn_model_gossipcop, X_train_gossipcop, X_test_gossipcop,␣
  ↪y_train_gossipcop, y_test_gossipcop, class_weights_dict_gossipcop,␣
  ↪early_stopping_gossipcop, reduce_lr, train_model=True)

# Train and evaluate the CNN model for PolitiFact dataset
cnn_model_politifact, y_pred_politifact, history_politifact =␣
  ↪train_and_evaluate(cnn_model_politifact, X_train_politifact,␣
  ↪X_test_politifact, y_train_politifact, y_test_politifact,␣
  ↪class_weights_dict_politifact, early_stopping_politifact, reduce_lr,␣
  ↪train_model=True)
```

```
Epoch 1/50
139/139             1s 9ms/step
 - val_f1: 0.8683 - val_precision: 0.7683 - val_recall: 0.9982
222/222             25s 92ms/step -
```

```
accuracy: 0.4799 - loss: 1.7763 - val_accuracy: 0.7671 - val_loss: 1.2929 -
learning_rate: 1.0000e-04
Epoch 2/50
139/139             1s 9ms/step
 - val_f1: 0.8825 - val_precision: 0.7996 - val_recall: 0.9847
222/222             18s 81ms/step -
accuracy: 0.6306 - loss: 1.2043 - val_accuracy: 0.8013 - val_loss: 0.9562 -
learning_rate: 1.0000e-04
Epoch 3/50
139/139             1s 9ms/step
 - val_f1: 0.8891 - val_precision: 0.8578 - val_recall: 0.9227
222/222             22s 89ms/step -
accuracy: 0.7706 - loss: 0.9178 - val_accuracy: 0.8236 - val_loss: 0.7375 -
learning_rate: 1.0000e-04
Epoch 4/50
139/139             1s 9ms/step
 - val_f1: 0.8701 - val_precision: 0.9047 - val_recall: 0.8380
222/222             20s 90ms/step -
accuracy: 0.8028 - loss: 0.7131 - val_accuracy: 0.7959 - val_loss: 0.5666 -
learning_rate: 1.0000e-04
Epoch 5/50
139/139             1s 9ms/step
 - val_f1: 0.8815 - val_precision: 0.9008 - val_recall: 0.8630
222/222             18s 82ms/step -
accuracy: 0.8223 - loss: 0.5492 - val_accuracy: 0.8148 - val_loss: 0.5063 -
learning_rate: 1.0000e-04
Epoch 6/50
139/139             2s 13ms/step
 - val_f1: 0.8900 - val_precision: 0.8979 - val_recall: 0.8822
222/222             23s 93ms/step -
accuracy: 0.8435 - loss: 0.4742 - val_accuracy: 0.8177 - val_loss: 0.4856 -
learning_rate: 1.0000e-04
Epoch 7/50
139/139             1s 9ms/step
 - val_f1: 0.8894 - val_precision: 0.8948 - val_recall: 0.8840
222/222             18s 83ms/step -
accuracy: 0.8602 - loss: 0.4300 - val_accuracy: 0.8171 - val_loss: 0.4837 -
learning_rate: 1.0000e-04
Epoch 8/50
139/139             1s 9ms/step
 - val_f1: 0.8889 - val_precision: 0.8933 - val_recall: 0.8846
222/222             20s 88ms/step -
accuracy: 0.8703 - loss: 0.4029 - val_accuracy: 0.8163 - val_loss: 0.4896 -
learning_rate: 1.0000e-04
Epoch 9/50
139/139             2s 13ms/step
 - val_f1: 0.8859 - val_precision: 0.8908 - val_recall: 0.8811
222/222             19s 87ms/step -
```

```
accuracy: 0.8789 - loss: 0.3717 - val_accuracy: 0.8168 - val_loss: 0.5024 -
learning_rate: 1.0000e-04
Epoch 10/50
139/139             1s 8ms/step
 - val_f1: 0.8905 - val_precision: 0.8856 - val_recall: 0.8955
222/222             21s 89ms/step -
accuracy: 0.8931 - loss: 0.3517 - val_accuracy: 0.8208 - val_loss: 0.5060 -
learning_rate: 1.0000e-04
139/139             1s 9ms/step
Accuracy: 0.8318
Precision: 0.8948
Recall: 0.8840
F1-score: 0.8894
ROC AUC: 0.8648
Confusion Matrix:
[[ 688  352]
 [ 393 2995]]
```





```
Epoch 1/50
7/7               0s 28ms/step
 - val_f1: 0.5520 - val_precision: 0.6703 - val_recall: 0.4692
11/11               4s 191ms/step -
accuracy: 0.5199 - loss: 3.4466 - val_accuracy: 0.5503 - val_loss: 3.3734 -
learning_rate: 1.0000e-04
Epoch 2/50
7/7               0s 23ms/step
 - val_f1: 0.4894 - val_precision: 0.7931 - val_recall: 0.3538
11/11               2s 208ms/step -
accuracy: 0.5408 - loss: 3.3440 - val_accuracy: 0.5740 - val_loss: 3.2736 -
learning_rate: 1.0000e-04
Epoch 3/50
7/7               0s 23ms/step
 - val_f1: 0.3636 - val_precision: 0.8571 - val_recall: 0.2308
11/11               3s 211ms/step -
accuracy: 0.5131 - loss: 3.2449 - val_accuracy: 0.4911 - val_loss: 3.1774 -
learning_rate: 1.0000e-04
Epoch 4/50
7/7               0s 17ms/step
```

```
 - val_f1: 0.3860 - val_precision: 0.8049 - val_recall: 0.2538
11/11            2s 137ms/step -
accuracy: 0.5551 - loss: 3.1478 - val_accuracy: 0.5089 - val_loss: 3.0846 -
learning_rate: 1.0000e-04
Epoch 5/50
7/7            0s 14ms/step
 - val_f1: 0.3473 - val_precision: 0.7838 - val_recall: 0.2231
11/11            3s 145ms/step -
accuracy: 0.5500 - loss: 3.0560 - val_accuracy: 0.4970 - val_loss: 2.9955 -
learning_rate: 1.0000e-04
Epoch 6/50
7/7            0s 14ms/step
 - val_f1: 0.3953 - val_precision: 0.8095 - val_recall: 0.2615
11/11            2s 150ms/step -
accuracy: 0.5314 - loss: 2.9663 - val_accuracy: 0.5089 - val_loss: 2.9097 -
learning_rate: 1.0000e-04
Epoch 7/50
7/7            0s 15ms/step
 - val_f1: 0.4444 - val_precision: 0.8000 - val_recall: 0.3077
11/11            2s 145ms/step -
accuracy: 0.5491 - loss: 2.8805 - val_accuracy: 0.5503 - val_loss: 2.8271 -
learning_rate: 1.0000e-04
Epoch 8/50
7/7            0s 31ms/step
 - val_f1: 0.4731 - val_precision: 0.7857 - val_recall: 0.3385
11/11            4s 245ms/step -
accuracy: 0.5716 - loss: 2.7985 - val_accuracy: 0.5740 - val_loss: 2.7477 -
learning_rate: 1.0000e-04
Epoch 9/50
7/7            0s 14ms/step
 - val_f1: 0.5026 - val_precision: 0.7869 - val_recall: 0.3692
11/11            4s 152ms/step -
accuracy: 0.5852 - loss: 2.7197 - val_accuracy: 0.5680 - val_loss: 2.6717 -
learning_rate: 1.0000e-04
Epoch 10/50
7/7            0s 15ms/step
 - val_f1: 0.5258 - val_precision: 0.7969 - val_recall: 0.3923
11/11            2s 134ms/step -
accuracy: 0.6081 - loss: 2.6439 - val_accuracy: 0.5740 - val_loss: 2.5987 -
learning_rate: 1.0000e-04
Epoch 11/50
7/7            0s 14ms/step
 - val_f1: 0.5381 - val_precision: 0.7910 - val_recall: 0.4077
11/11            1s 135ms/step -
accuracy: 0.5931 - loss: 2.5709 - val_accuracy: 0.5799 - val_loss: 2.5284 -
learning_rate: 1.0000e-04
Epoch 12/50
7/7            0s 15ms/step
```

```
 - val_f1: 0.5381 - val_precision: 0.7910 - val_recall: 0.4077
11/11            3s 142ms/step -
accuracy: 0.6228 - loss: 2.5012 - val_accuracy: 0.5799 - val_loss: 2.4610 -
learning_rate: 1.0000e-04
Epoch 13/50
7/7            0s 14ms/step
 - val_f1: 0.5528 - val_precision: 0.7971 - val_recall: 0.4231
11/11            2s 141ms/step -
accuracy: 0.6171 - loss: 2.4339 - val_accuracy: 0.5799 - val_loss: 2.3962 -
learning_rate: 1.0000e-04
Epoch 14/50
7/7            0s 23ms/step
 - val_f1: 0.5528 - val_precision: 0.7971 - val_recall: 0.4231
11/11            2s 153ms/step -
accuracy: 0.6070 - loss: 2.3687 - val_accuracy: 0.5799 - val_loss: 2.3341 -
learning_rate: 1.0000e-04
Epoch 15/50
7/7            0s 24ms/step
 - val_f1: 0.5672 - val_precision: 0.8028 - val_recall: 0.4385
11/11            2s 208ms/step -
accuracy: 0.6136 - loss: 2.3062 - val_accuracy: 0.5799 - val_loss: 2.2742 -
learning_rate: 1.0000e-04
Epoch 16/50
7/7            0s 24ms/step
 - val_f1: 0.5854 - val_precision: 0.8000 - val_recall: 0.4615
11/11            3s 227ms/step -
accuracy: 0.6062 - loss: 2.2483 - val_accuracy: 0.5976 - val_loss: 2.2164 -
learning_rate: 1.0000e-04
Epoch 17/50
7/7            0s 14ms/step
 - val_f1: 0.5813 - val_precision: 0.8082 - val_recall: 0.4538
11/11            2s 145ms/step -
accuracy: 0.6588 - loss: 2.1895 - val_accuracy: 0.5976 - val_loss: 2.1613 -
learning_rate: 1.0000e-04
Epoch 18/50
7/7            0s 21ms/step
 - val_f1: 0.5854 - val_precision: 0.8000 - val_recall: 0.4615
11/11            3s 154ms/step -
accuracy: 0.6299 - loss: 2.1349 - val_accuracy: 0.5976 - val_loss: 2.1081 -
learning_rate: 1.0000e-04
Epoch 19/50
7/7            0s 16ms/step
 - val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11            2s 136ms/step -
accuracy: 0.6777 - loss: 2.0810 - val_accuracy: 0.5976 - val_loss: 2.0572 -
learning_rate: 1.0000e-04
Epoch 20/50
7/7            0s 14ms/step
```

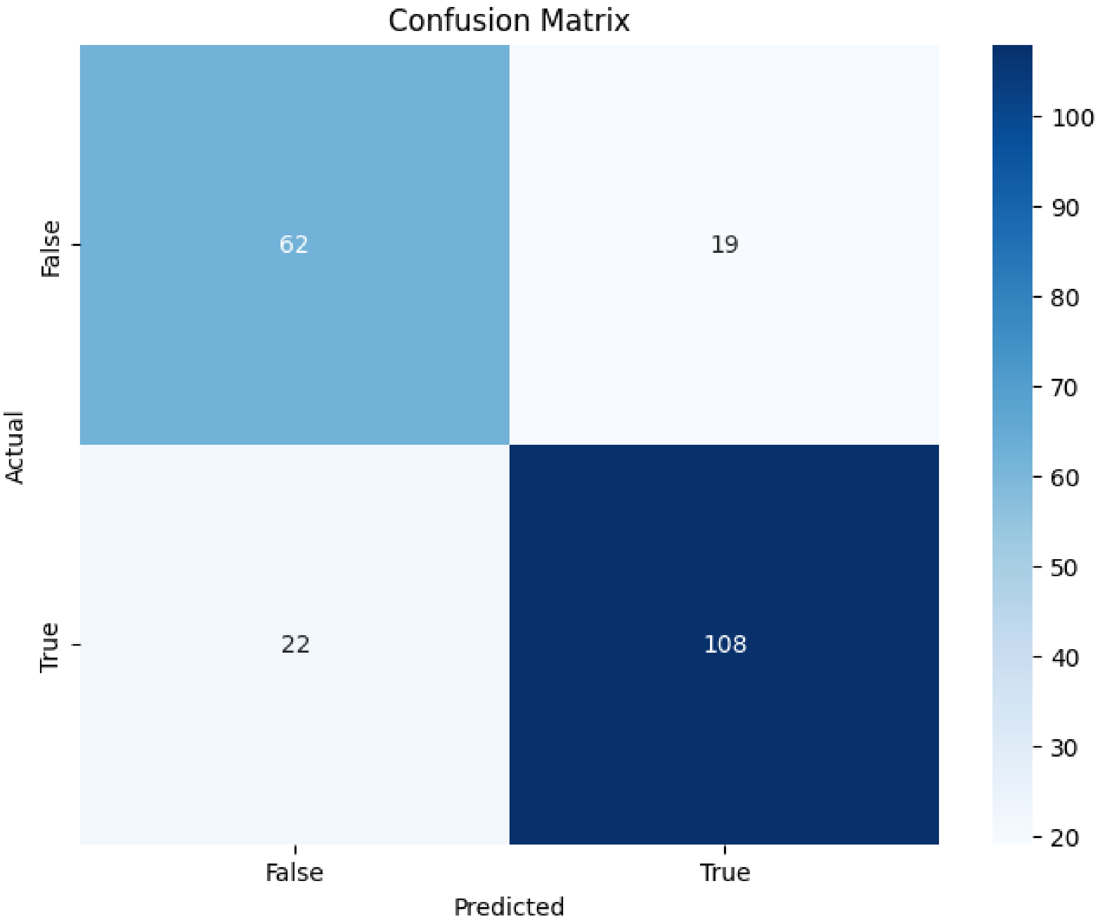
```
 - val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11            3s 137ms/step -
accuracy: 0.6405 - loss: 2.0301 - val_accuracy: 0.5976 - val_loss: 2.0084 -
learning_rate: 1.0000e-04
Epoch 21/50
7/7            0s 26ms/step
 - val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11            2s 188ms/step -
accuracy: 0.6611 - loss: 1.9806 - val_accuracy: 0.5976 - val_loss: 1.9612 -
learning_rate: 1.0000e-04
Epoch 22/50
7/7            0s 24ms/step
 - val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11            2s 220ms/step -
accuracy: 0.6648 - loss: 1.9326 - val_accuracy: 0.5976 - val_loss: 1.9161 -
learning_rate: 1.0000e-04
Epoch 23/50
7/7            0s 14ms/step
 - val_f1: 0.5962 - val_precision: 0.7949 - val_recall: 0.4769
11/11            2s 170ms/step -
accuracy: 0.6433 - loss: 1.8880 - val_accuracy: 0.5917 - val_loss: 1.8723 -
learning_rate: 1.0000e-04
Epoch 24/50
7/7            0s 16ms/step
 - val_f1: 0.6029 - val_precision: 0.7975 - val_recall: 0.4846
11/11            2s 134ms/step -
accuracy: 0.6625 - loss: 1.8476 - val_accuracy: 0.5917 - val_loss: 1.8307 -
learning_rate: 1.0000e-04
Epoch 25/50
7/7            0s 14ms/step
 - val_f1: 0.6029 - val_precision: 0.7975 - val_recall: 0.4846
11/11            3s 145ms/step -
accuracy: 0.6726 - loss: 1.8032 - val_accuracy: 0.5976 - val_loss: 1.7905 -
learning_rate: 1.0000e-04
Epoch 26/50
7/7            0s 15ms/step
 - val_f1: 0.5894 - val_precision: 0.7922 - val_recall: 0.4692
11/11            2s 139ms/step -
accuracy: 0.6613 - loss: 1.7615 - val_accuracy: 0.5917 - val_loss: 1.7524 -
learning_rate: 1.0000e-04
Epoch 27/50
7/7            0s 14ms/step
 - val_f1: 0.5894 - val_precision: 0.7922 - val_recall: 0.4692
11/11            2s 138ms/step -
accuracy: 0.6430 - loss: 1.7235 - val_accuracy: 0.5917 - val_loss: 1.7154 -
learning_rate: 1.0000e-04
Epoch 28/50
7/7            0s 24ms/step
```

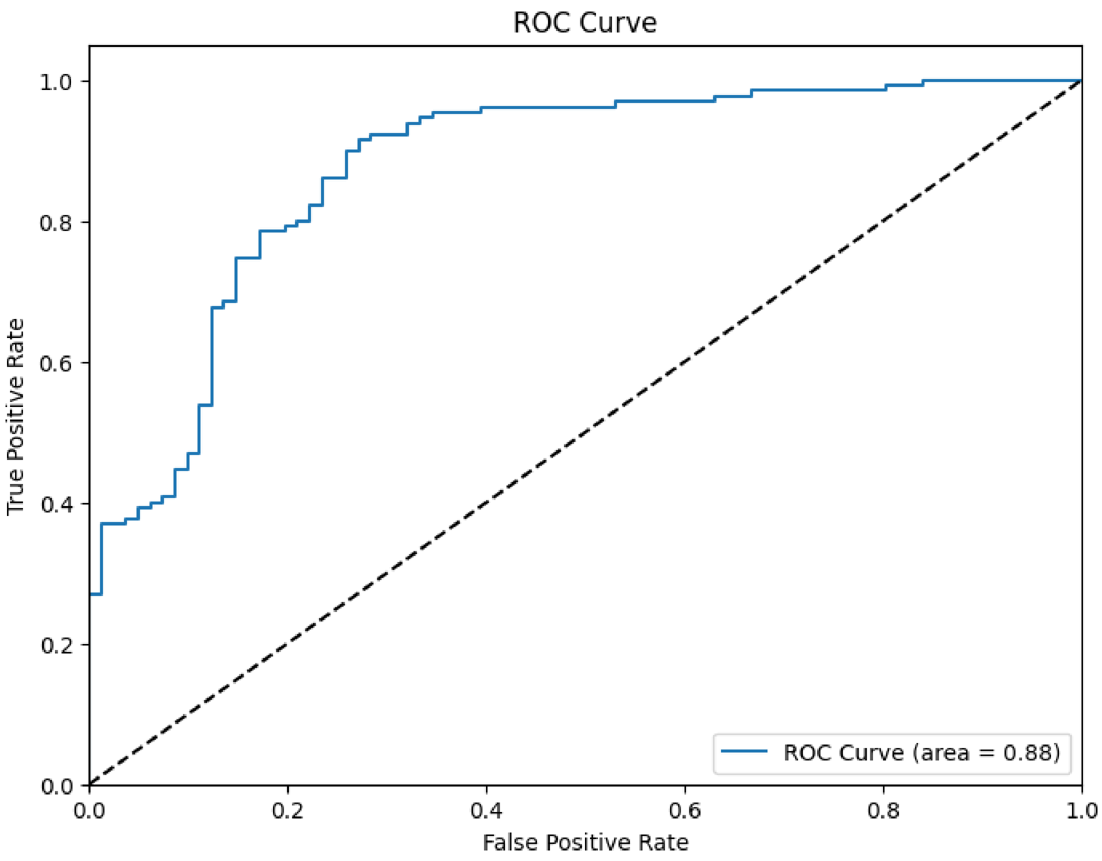
```
 - val_f1: 0.6226 - val_precision: 0.8049 - val_recall: 0.5077
11/11            2s 198ms/step -
accuracy: 0.6493 - loss: 1.6853 - val_accuracy: 0.6036 - val_loss: 1.6796 -
learning_rate: 1.0000e-04
Epoch 29/50
7/7            0s 21ms/step
 - val_f1: 0.6095 - val_precision: 0.8000 - val_recall: 0.4923
11/11            3s 222ms/step -
accuracy: 0.6859 - loss: 1.6486 - val_accuracy: 0.6036 - val_loss: 1.6457 -
learning_rate: 1.0000e-04
Epoch 30/50
7/7            0s 15ms/step
 - val_f1: 0.6226 - val_precision: 0.8049 - val_recall: 0.5077
11/11            2s 143ms/step -
accuracy: 0.6709 - loss: 1.6135 - val_accuracy: 0.6036 - val_loss: 1.6128 -
learning_rate: 1.0000e-04
Epoch 31/50
7/7            0s 15ms/step
 - val_f1: 0.6161 - val_precision: 0.8025 - val_recall: 0.5000
11/11            2s 139ms/step -
accuracy: 0.6866 - loss: 1.5772 - val_accuracy: 0.6036 - val_loss: 1.5815 -
learning_rate: 1.0000e-04
Epoch 32/50
7/7            0s 14ms/step
 - val_f1: 0.6355 - val_precision: 0.8095 - val_recall: 0.5231
11/11            3s 138ms/step -
accuracy: 0.6750 - loss: 1.5444 - val_accuracy: 0.6213 - val_loss: 1.5508 -
learning_rate: 1.0000e-04
Epoch 33/50
7/7            0s 15ms/step
 - val_f1: 0.6544 - val_precision: 0.8161 - val_recall: 0.5462
11/11            3s 137ms/step -
accuracy: 0.6792 - loss: 1.5118 - val_accuracy: 0.6154 - val_loss: 1.5212 -
learning_rate: 1.0000e-04
Epoch 34/50
7/7            0s 23ms/step
 - val_f1: 0.6481 - val_precision: 0.8140 - val_recall: 0.5385
11/11            2s 177ms/step -
accuracy: 0.6978 - loss: 1.4777 - val_accuracy: 0.6154 - val_loss: 1.4930 -
learning_rate: 1.0000e-04
Epoch 35/50
7/7            0s 24ms/step
 - val_f1: 0.6574 - val_precision: 0.8256 - val_recall: 0.5462
11/11            2s 211ms/step -
accuracy: 0.7131 - loss: 1.4434 - val_accuracy: 0.6213 - val_loss: 1.4653 -
learning_rate: 1.0000e-04
Epoch 36/50
7/7            0s 14ms/step
```

```
 - val_f1: 0.6758 - val_precision: 0.8315 - val_recall: 0.5692
11/11            2s 187ms/step -
accuracy: 0.7246 - loss: 1.4081 - val_accuracy: 0.6272 - val_loss: 1.4376 -
learning_rate: 1.0000e-04
Epoch 37/50
7/7            0s 16ms/step
 - val_f1: 0.7168 - val_precision: 0.8438 - val_recall: 0.6231
11/11            2s 138ms/step -
accuracy: 0.7385 - loss: 1.3795 - val_accuracy: 0.6686 - val_loss: 1.4098 -
learning_rate: 1.0000e-04
Epoch 38/50
7/7            0s 14ms/step
 - val_f1: 0.7225 - val_precision: 0.8454 - val_recall: 0.6308
11/11            3s 144ms/step -
accuracy: 0.7841 - loss: 1.3374 - val_accuracy: 0.6805 - val_loss: 1.3829 -
learning_rate: 1.0000e-04
Epoch 39/50
7/7            0s 14ms/step
 - val_f1: 0.7225 - val_precision: 0.8454 - val_recall: 0.6308
11/11            3s 142ms/step -
accuracy: 0.8081 - loss: 1.2946 - val_accuracy: 0.6923 - val_loss: 1.3548 -
learning_rate: 1.0000e-04
Epoch 40/50
7/7            0s 14ms/step
 - val_f1: 0.7336 - val_precision: 0.8485 - val_recall: 0.6462
11/11            2s 135ms/step -
accuracy: 0.8427 - loss: 1.2564 - val_accuracy: 0.6982 - val_loss: 1.3299 -
learning_rate: 1.0000e-04
Epoch 41/50
7/7            0s 25ms/step
 - val_f1: 0.7731 - val_precision: 0.8519 - val_recall: 0.7077
11/11            2s 195ms/step -
accuracy: 0.8446 - loss: 1.2114 - val_accuracy: 0.7160 - val_loss: 1.2994 -
learning_rate: 1.0000e-04
Epoch 42/50
7/7            0s 23ms/step
 - val_f1: 0.7884 - val_precision: 0.8559 - val_recall: 0.7308
11/11            3s 224ms/step -
accuracy: 0.8655 - loss: 1.1665 - val_accuracy: 0.7219 - val_loss: 1.2715 -
learning_rate: 1.0000e-04
Epoch 43/50
7/7            0s 17ms/step
 - val_f1: 0.8130 - val_precision: 0.8621 - val_recall: 0.7692
11/11            2s 138ms/step -
accuracy: 0.8877 - loss: 1.1128 - val_accuracy: 0.7456 - val_loss: 1.2401 -
learning_rate: 1.0000e-04
Epoch 44/50
7/7            0s 15ms/step
```

```
 - val_f1: 0.8379 - val_precision: 0.8618 - val_recall: 0.8154
11/11             3s 145ms/step -
accuracy: 0.9164 - loss: 1.0637 - val_accuracy: 0.7574 - val_loss: 1.2110 -
learning_rate: 1.0000e-04
Epoch 45/50
7/7             0s 16ms/step
 - val_f1: 0.8392 - val_precision: 0.8560 - val_recall: 0.8231
11/11             2s 143ms/step -
accuracy: 0.9272 - loss: 1.0242 - val_accuracy: 0.7574 - val_loss: 1.1850 -
learning_rate: 1.0000e-04
Epoch 46/50
7/7             0s 15ms/step
 - val_f1: 0.8300 - val_precision: 0.8537 - val_recall: 0.8077
11/11             2s 139ms/step -
accuracy: 0.9338 - loss: 0.9899 - val_accuracy: 0.7574 - val_loss: 1.1690 -
learning_rate: 1.0000e-04
Epoch 47/50
7/7             0s 16ms/step
 - val_f1: 0.8392 - val_precision: 0.8560 - val_recall: 0.8231
11/11             3s 138ms/step -
accuracy: 0.9411 - loss: 0.9439 - val_accuracy: 0.7633 - val_loss: 1.1432 -
learning_rate: 1.0000e-04
Epoch 48/50
7/7             0s 26ms/step
 - val_f1: 0.8346 - val_precision: 0.8548 - val_recall: 0.8154
11/11             3s 222ms/step -
accuracy: 0.9452 - loss: 0.9129 - val_accuracy: 0.7633 - val_loss: 1.1298 -
learning_rate: 1.0000e-04
Epoch 49/50
7/7             0s 15ms/step
 - val_f1: 0.8359 - val_precision: 0.8492 - val_recall: 0.8231
11/11             2s 179ms/step -
accuracy: 0.9453 - loss: 0.8968 - val_accuracy: 0.7692 - val_loss: 1.1144 -
learning_rate: 1.0000e-04
Epoch 50/50
7/7             0s 17ms/step
 - val_f1: 0.8405 - val_precision: 0.8504 - val_recall: 0.8308
11/11             2s 137ms/step -
accuracy: 0.9462 - loss: 0.8765 - val_accuracy: 0.7811 - val_loss: 1.1007 -
learning_rate: 1.0000e-04
7/7             0s 14ms/step
Accuracy: 0.8057
Precision: 0.8504
Recall: 0.8308
F1-score: 0.8405
ROC AUC: 0.8760
Confusion Matrix:
[[ 62  19]
```

```
[ 22 108]]
```


Confusion Matrix
Actual
False
True
Predicted
False
True
62
19
22
108
100
90
80
70
60
50
40
30
20

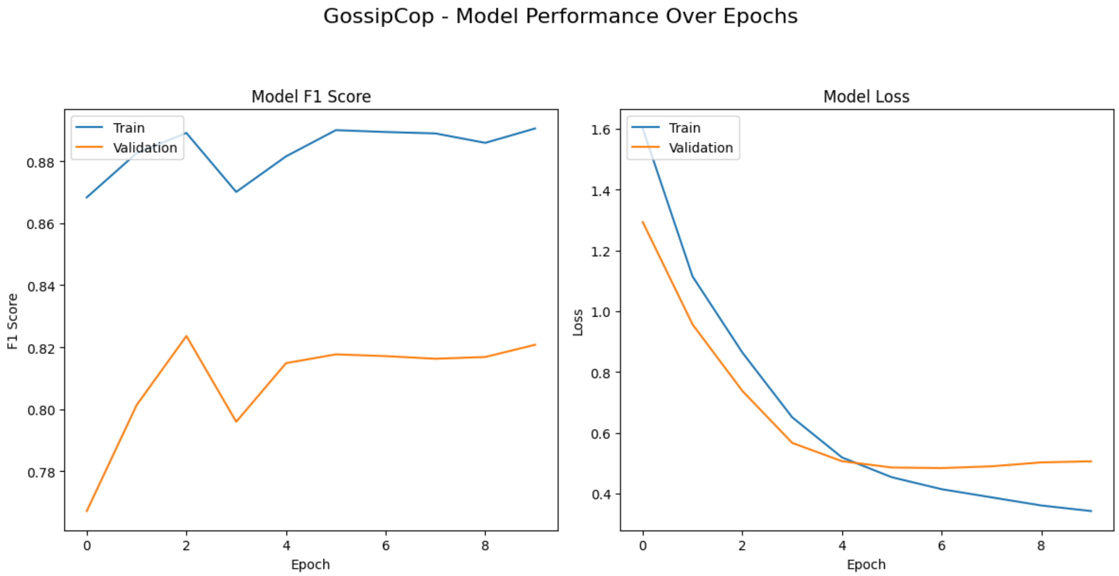


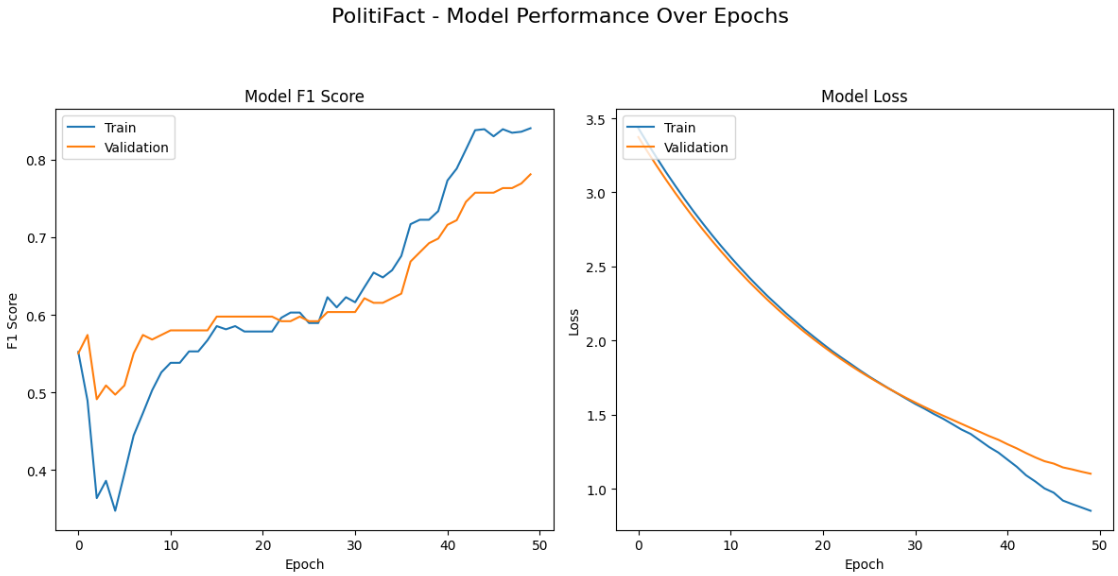
1.8.2 Model Performance over Epochs

The training history including F1 Score and loss for both training and validation sets is plotted for each dataset.

[14]:
```
def plot_training_history(history, dataset_name):
    # Plot training & validation F1 score values
    plt.figure(figsize=(12, 6))
    plt.suptitle(f'{dataset_name} - Model Performance Over Epochs',␣
 ↪fontsize=16, y=1.02)  # y parameter for spacing

    plt.subplot(1, 2, 1)
    if 'val_f1_score' in history.history:
        plt.plot(history.history['val_f1_score'])
        plt.title('Model F1 Score')
        plt.ylabel('F1 Score')
    else:
        plt.plot(history.history['accuracy'])
        plt.title('Model Accuracy')
        plt.ylabel('Accuracy')
```

```
    plt.plot(history.history['val_accuracy'])
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Validation'], loc='upper left')

    # Plot training & validation loss values
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model Loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Validation'], loc='upper left')

    plt.tight_layout(rect=[0, 0, 1, 0.96])
    plt.show()

# Plot training history for GossipCop
plot_training_history(history_gossipcop, "GossipCop")

# Plot training history for PolitiFact
plot_training_history(history_politifact, "PolitiFact")
```





F1 score graphs show an increase over the epochs. It means the models are learning and improving their ability to balance precision and recall.

The loss graphs show a steady decrease in both training and validation loss at the beginning, that stabilizes towards the later epochs. This means the models are converging and learning effectively without overfitting.

1.8.3 Comparison of my results with baseline Shu et al. [7]

Shu et al. Baseline Results:

```
GossipCop
    Accuracy: 0.723
    Precision: 0.751
    Recall: 0.701
    F1: 0.725

PolitiFact
    Accuracy: 0.629
    Precision: 0.807
    Recall: 0.456
    F1: 0.583
```

My Model Results:

```
GossipCop
    Accuracy: 0.8318
    Precision: 0.8948
    Recall: 0.8840
    F1: 0.8894
```

```
PolitiFact
    Accuracy: 0.8057
    Precision: 0.8504
    Recall: 0.8308
    F1: 0.8405
```

My models significantly outperformed Shu et al.'s CNN results in all metrics for both datasets.

1.9 Implement 5-Fold Cross-Validation

I will randomly split the datasets into five parts and conduct 5-fold cross-validation to obtain robust results, as Bian et al.[5] did. Cross-validation is a statistical method used to estimate the performance of machine learning models. As Browne et al.[15] explain: "In its simplest form, the leaving one out at a time method, this involves partitioning a sample of size N into a calibration sample of size N-1 and a validation sample of size 1 and repeating the process N times. An average of the N cross-validation index values is then used." This method involves splitting the data into a number of subsets (folds), training the model on some subsets while testing it on the remaining subset, and repeating this process several times. The performance metrics are then averaged over all iterations to provide a more robust evaluation.

1.9.1 Cross-Validation Function

I could not use scikeras for k-fold cross validation due to compatibility issues with existing models, so I implemented k-fold cross validation manually.

[15]:

```
# Function to perform cross-validation
def run_kfold_cross_validation(X_data, y_data, create_model_func, dataset_name,␣
 ↪n_splits=5):
    X = np.array(X_data)
    y = np.array(y_data)

    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
    fold_no = 1

    # Initialize lists to store fold-wise performance metrics
    f1_scores, precision_scores, recall_scores = [], [], []

    for train_index, test_index in kf.split(X):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]

        # Clearing the TensorFlow backend to prevent data leakage
        clear_session()

        # Create a new instance of the CNN model
        model = create_model_func(input_length)

        # Train the model
        print(f'Training for fold {fold_no} on {dataset_name}...')
```

```
        model.fit(X_train, y_train, epochs=50, batch_size=64,␣
 ↪validation_data=(X_test, y_test))

        # Evaluate the model
        y_pred = model.predict(X_test)
        y_pred = (y_pred > 0.5).astype(int)

        # Calculate metrics
        f1 = f1_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred)

        f1_scores.append(f1)
        precision_scores.append(precision)
        recall_scores.append(recall)

        print(f'Fold {fold_no} - Precision: {precision:.4f}, Recall: {recall:.
 ↪4f}, F1 Score: {f1:.4f}')

        fold_no += 1

    # Return the average scores
    print(f'Average Precision: {np.mean(precision_scores):.4f}')
    print(f'Average Recall: {np.mean(recall_scores):.4f}')
    print(f'Average F1 Score: {np.mean(f1_scores):.4f}')
```

1.9.2 Cross-Validation Scores

[16]:

```
# Run cross-validation
run_kfold_cross_validation(X_gossipcop, y_gossipcop,␣
 ↪create_cnn_model_gossipcop, "GossipCop")
run_kfold_cross_validation(X_politifact, y_politifact,␣
 ↪create_cnn_model_politifact, "PolitiFact")
```

```
Training for fold 1 on GossipCop…
Epoch 1/50
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
277/277                   26s 84ms/step -
accuracy: 0.7266 - loss: 1.6738 - val_accuracy: 0.7651 - val_loss: 1.0707
Epoch 2/50
277/277                   41s 84ms/step -
accuracy: 0.7560 - loss: 0.9933 - val_accuracy: 0.7651 - val_loss: 0.7552
Epoch 3/50
277/277                   38s 73ms/step -
```

```
accuracy: 0.7560 - loss: 0.7377 - val_accuracy: 0.7651 - val_loss: 0.6311
Epoch 4/50
277/277            23s 83ms/step -
accuracy: 0.7560 - loss: 0.6369 - val_accuracy: 0.7651 - val_loss: 0.5842
Epoch 5/50
277/277            39s 75ms/step -
accuracy: 0.7560 - loss: 0.5990 - val_accuracy: 0.7651 - val_loss: 0.5660
Epoch 6/50
277/277            41s 74ms/step -
accuracy: 0.7560 - loss: 0.5845 - val_accuracy: 0.7651 - val_loss: 0.5587
Epoch 7/50
277/277            41s 74ms/step -
accuracy: 0.7560 - loss: 0.5756 - val_accuracy: 0.7651 - val_loss: 0.5547
Epoch 8/50
277/277            43s 83ms/step -
accuracy: 0.7560 - loss: 0.5718 - val_accuracy: 0.7651 - val_loss: 0.5525
Epoch 9/50
277/277            38s 73ms/step -
accuracy: 0.7560 - loss: 0.5728 - val_accuracy: 0.7651 - val_loss: 0.5487
Epoch 10/50
277/277            22s 78ms/step -
accuracy: 0.7560 - loss: 0.5636 - val_accuracy: 0.7651 - val_loss: 0.5128
Epoch 11/50
277/277            42s 81ms/step -
accuracy: 0.7635 - loss: 0.5077 - val_accuracy: 0.7949 - val_loss: 0.4691
Epoch 12/50
277/277            39s 74ms/step -
accuracy: 0.7997 - loss: 0.4500 - val_accuracy: 0.8071 - val_loss: 0.4605
Epoch 13/50
277/277            41s 74ms/step -
accuracy: 0.8238 - loss: 0.4204 - val_accuracy: 0.8119 - val_loss: 0.4585
Epoch 14/50
277/277            23s 81ms/step -
accuracy: 0.8396 - loss: 0.4005 - val_accuracy: 0.8132 - val_loss: 0.4558
Epoch 15/50
277/277            39s 75ms/step -
accuracy: 0.8474 - loss: 0.3851 - val_accuracy: 0.8171 - val_loss: 0.4543
Epoch 16/50
277/277            41s 75ms/step -
accuracy: 0.8584 - loss: 0.3723 - val_accuracy: 0.8209 - val_loss: 0.4517
Epoch 17/50
277/277            22s 79ms/step -
accuracy: 0.8674 - loss: 0.3565 - val_accuracy: 0.8238 - val_loss: 0.4459
Epoch 18/50
277/277            41s 78ms/step -
accuracy: 0.8798 - loss: 0.3394 - val_accuracy: 0.8290 - val_loss: 0.4453
Epoch 19/50
277/277            40s 75ms/step -
```

```
accuracy: 0.8845 - loss: 0.3280 - val_accuracy: 0.8304 - val_loss: 0.4481
Epoch 20/50
277/277            43s 82ms/step -
accuracy: 0.8872 - loss: 0.3182 - val_accuracy: 0.8299 - val_loss: 0.4523
Epoch 21/50
277/277            22s 78ms/step -
accuracy: 0.8909 - loss: 0.3118 - val_accuracy: 0.8304 - val_loss: 0.4529
Epoch 22/50
277/277            42s 82ms/step -
accuracy: 0.9004 - loss: 0.2969 - val_accuracy: 0.8324 - val_loss: 0.4599
Epoch 23/50
277/277            39s 75ms/step -
accuracy: 0.9034 - loss: 0.2929 - val_accuracy: 0.8338 - val_loss: 0.4656
Epoch 24/50
277/277            45s 89ms/step -
accuracy: 0.9089 - loss: 0.2806 - val_accuracy: 0.8324 - val_loss: 0.4747
Epoch 25/50
277/277            37s 75ms/step -
accuracy: 0.9109 - loss: 0.2756 - val_accuracy: 0.8308 - val_loss: 0.4824
Epoch 26/50
277/277            41s 74ms/step -
accuracy: 0.9161 - loss: 0.2683 - val_accuracy: 0.8315 - val_loss: 0.4905
Epoch 27/50
277/277            22s 80ms/step -
accuracy: 0.9216 - loss: 0.2628 - val_accuracy: 0.8297 - val_loss: 0.4984
Epoch 28/50
277/277            41s 79ms/step -
accuracy: 0.9216 - loss: 0.2533 - val_accuracy: 0.8299 - val_loss: 0.5090
Epoch 29/50
277/277            23s 84ms/step -
accuracy: 0.9263 - loss: 0.2475 - val_accuracy: 0.8308 - val_loss: 0.5177
Epoch 30/50
277/277            40s 80ms/step -
accuracy: 0.9290 - loss: 0.2375 - val_accuracy: 0.8306 - val_loss: 0.5291
Epoch 31/50
277/277            40s 75ms/step -
accuracy: 0.9355 - loss: 0.2351 - val_accuracy: 0.8315 - val_loss: 0.5367
Epoch 32/50
277/277            41s 74ms/step -
accuracy: 0.9368 - loss: 0.2281 - val_accuracy: 0.8324 - val_loss: 0.5467
Epoch 33/50
277/277            43s 82ms/step -
accuracy: 0.9399 - loss: 0.2172 - val_accuracy: 0.8318 - val_loss: 0.5618
Epoch 34/50
277/277            39s 74ms/step -
accuracy: 0.9430 - loss: 0.2155 - val_accuracy: 0.8327 - val_loss: 0.5686
Epoch 35/50
277/277            41s 73ms/step -
```

```
accuracy: 0.9445 - loss: 0.2064 - val_accuracy: 0.8322 - val_loss: 0.5795
Epoch 36/50
277/277             22s 79ms/step -
accuracy: 0.9484 - loss: 0.2031 - val_accuracy: 0.8308 - val_loss: 0.5969
Epoch 37/50
277/277             21s 75ms/step -
accuracy: 0.9521 - loss: 0.1953 - val_accuracy: 0.8315 - val_loss: 0.5999
Epoch 38/50
277/277             43s 83ms/step -
accuracy: 0.9520 - loss: 0.1918 - val_accuracy: 0.8311 - val_loss: 0.6080
Epoch 39/50
277/277             40s 80ms/step -
accuracy: 0.9543 - loss: 0.1853 - val_accuracy: 0.8295 - val_loss: 0.6240
Epoch 40/50
277/277             20s 73ms/step -
accuracy: 0.9564 - loss: 0.1782 - val_accuracy: 0.8277 - val_loss: 0.6316
Epoch 41/50
277/277             21s 76ms/step -
accuracy: 0.9594 - loss: 0.1726 - val_accuracy: 0.8290 - val_loss: 0.6525
Epoch 42/50
277/277             40s 74ms/step -
accuracy: 0.9607 - loss: 0.1723 - val_accuracy: 0.8272 - val_loss: 0.6576
Epoch 43/50
277/277             42s 79ms/step -
accuracy: 0.9628 - loss: 0.1640 - val_accuracy: 0.8238 - val_loss: 0.6661
Epoch 44/50
277/277             39s 73ms/step -
accuracy: 0.9633 - loss: 0.1610 - val_accuracy: 0.8241 - val_loss: 0.6811
Epoch 45/50
277/277             22s 79ms/step -
accuracy: 0.9661 - loss: 0.1555 - val_accuracy: 0.8229 - val_loss: 0.6887
Epoch 46/50
277/277             40s 77ms/step -
accuracy: 0.9695 - loss: 0.1486 - val_accuracy: 0.8254 - val_loss: 0.7002
Epoch 47/50
277/277             42s 79ms/step -
accuracy: 0.9696 - loss: 0.1457 - val_accuracy: 0.8243 - val_loss: 0.7157
Epoch 48/50
277/277             40s 74ms/step -
accuracy: 0.9687 - loss: 0.1443 - val_accuracy: 0.8254 - val_loss: 0.7186
Epoch 49/50
277/277             41s 74ms/step -
accuracy: 0.9716 - loss: 0.1415 - val_accuracy: 0.8245 - val_loss: 0.7283
Epoch 50/50
277/277             22s 80ms/step -
accuracy: 0.9708 - loss: 0.1376 - val_accuracy: 0.8245 - val_loss: 0.7521
139/139             1s 9ms/step
Fold 1 - Precision: 0.8644, Recall: 0.9141, F1 Score: 0.8885
```

```
Training for fold 2 on GossipCop…
Epoch 1/50
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
277/277            23s 75ms/step -
accuracy: 0.6808 - loss: 1.6950 - val_accuracy: 0.7565 - val_loss: 1.0686
Epoch 2/50
277/277            43s 82ms/step -
accuracy: 0.7576 - loss: 0.9791 - val_accuracy: 0.7565 - val_loss: 0.7532
Epoch 3/50
277/277            38s 73ms/step -
accuracy: 0.7576 - loss: 0.7269 - val_accuracy: 0.7565 - val_loss: 0.6334
Epoch 4/50
277/277            22s 80ms/step -
accuracy: 0.7576 - loss: 0.6298 - val_accuracy: 0.7565 - val_loss: 0.5893
Epoch 5/50
277/277            40s 75ms/step -
accuracy: 0.7576 - loss: 0.5950 - val_accuracy: 0.7565 - val_loss: 0.5732
Epoch 6/50
277/277            41s 74ms/step -
accuracy: 0.7576 - loss: 0.5815 - val_accuracy: 0.7565 - val_loss: 0.5667
Epoch 7/50
277/277            41s 74ms/step -
accuracy: 0.7576 - loss: 0.5756 - val_accuracy: 0.7565 - val_loss: 0.5639
Epoch 8/50
277/277            41s 74ms/step -
accuracy: 0.7576 - loss: 0.5712 - val_accuracy: 0.7565 - val_loss: 0.5621
Epoch 9/50
277/277            22s 80ms/step -
accuracy: 0.7576 - loss: 0.5701 - val_accuracy: 0.7565 - val_loss: 0.5611
Epoch 10/50
277/277            41s 81ms/step -
accuracy: 0.7576 - loss: 0.5717 - val_accuracy: 0.7565 - val_loss: 0.5602
Epoch 11/50
277/277            43s 88ms/step -
accuracy: 0.7576 - loss: 0.5672 - val_accuracy: 0.7565 - val_loss: 0.5596
Epoch 12/50
277/277            37s 74ms/step -
accuracy: 0.7576 - loss: 0.5661 - val_accuracy: 0.7565 - val_loss: 0.5592
Epoch 13/50
277/277            22s 80ms/step -
accuracy: 0.7576 - loss: 0.5683 - val_accuracy: 0.7565 - val_loss: 0.5586
Epoch 14/50
277/277            40s 76ms/step -
accuracy: 0.7576 - loss: 0.5661 - val_accuracy: 0.7565 - val_loss: 0.5583
Epoch 15/50
```

```
277/277             21s 77ms/step -
accuracy: 0.7576 - loss: 0.5662 - val_accuracy: 0.7565 - val_loss: 0.5579
Epoch 16/50
277/277             44s 87ms/step -
accuracy: 0.7576 - loss: 0.5670 - val_accuracy: 0.7565 - val_loss: 0.5574
Epoch 17/50
277/277             37s 74ms/step -
accuracy: 0.7576 - loss: 0.5665 - val_accuracy: 0.7565 - val_loss: 0.5561
Epoch 18/50
277/277             41s 75ms/step -
accuracy: 0.7576 - loss: 0.5619 - val_accuracy: 0.7565 - val_loss: 0.5213
Epoch 19/50
277/277             22s 80ms/step -
accuracy: 0.7576 - loss: 0.5053 - val_accuracy: 0.7565 - val_loss: 0.4764
Epoch 20/50
277/277             41s 82ms/step -
accuracy: 0.7636 - loss: 0.4570 - val_accuracy: 0.7565 - val_loss: 0.4651
Epoch 21/50
277/277             23s 81ms/step -
accuracy: 0.7916 - loss: 0.4346 - val_accuracy: 0.7565 - val_loss: 0.4602
Epoch 22/50
277/277             41s 80ms/step -
accuracy: 0.7908 - loss: 0.4222 - val_accuracy: 0.8130 - val_loss: 0.4570
Epoch 23/50
277/277             20s 73ms/step -
accuracy: 0.7993 - loss: 0.4103 - val_accuracy: 0.8187 - val_loss: 0.4547
Epoch 24/50
277/277             21s 76ms/step -
accuracy: 0.8050 - loss: 0.3987 - val_accuracy: 0.8198 - val_loss: 0.4538
Epoch 25/50
277/277             40s 74ms/step -
accuracy: 0.8196 - loss: 0.3871 - val_accuracy: 0.8214 - val_loss: 0.4534
Epoch 26/50
277/277             22s 80ms/step -
accuracy: 0.8300 - loss: 0.3747 - val_accuracy: 0.8263 - val_loss: 0.4513
Epoch 27/50
277/277             39s 73ms/step -
accuracy: 0.8361 - loss: 0.3655 - val_accuracy: 0.8275 - val_loss: 0.4518
Epoch 28/50
277/277             22s 80ms/step -
accuracy: 0.8388 - loss: 0.3544 - val_accuracy: 0.8308 - val_loss: 0.4507
Epoch 29/50
277/277             40s 78ms/step -
accuracy: 0.8411 - loss: 0.3475 - val_accuracy: 0.8313 - val_loss: 0.4516
Epoch 30/50
277/277             40s 75ms/step -
accuracy: 0.8461 - loss: 0.3377 - val_accuracy: 0.8311 - val_loss: 0.4554
Epoch 31/50
```

```
277/277             44s 88ms/step -
accuracy: 0.8527 - loss: 0.3307 - val_accuracy: 0.8288 - val_loss: 0.4597
Epoch 32/50
277/277             37s 74ms/step -
accuracy: 0.8528 - loss: 0.3282 - val_accuracy: 0.8299 - val_loss: 0.4613
Epoch 33/50
277/277             22s 80ms/step -
accuracy: 0.8566 - loss: 0.3204 - val_accuracy: 0.8284 - val_loss: 0.4623
Epoch 34/50
277/277             41s 79ms/step -
accuracy: 0.8587 - loss: 0.3135 - val_accuracy: 0.8270 - val_loss: 0.4696
Epoch 35/50
277/277             40s 74ms/step -
accuracy: 0.8840 - loss: 0.3032 - val_accuracy: 0.8263 - val_loss: 0.4754
Epoch 36/50
277/277             41s 75ms/step -
accuracy: 0.8881 - loss: 0.3043 - val_accuracy: 0.8268 - val_loss: 0.4816
Epoch 37/50
277/277             41s 74ms/step -
accuracy: 0.8930 - loss: 0.2959 - val_accuracy: 0.8252 - val_loss: 0.4857
Epoch 38/50
277/277             41s 74ms/step -
accuracy: 0.8942 - loss: 0.2944 - val_accuracy: 0.8257 - val_loss: 0.4946
Epoch 39/50
277/277             41s 73ms/step -
accuracy: 0.9009 - loss: 0.2876 - val_accuracy: 0.8229 - val_loss: 0.5001
Epoch 40/50
277/277             24s 87ms/step -
accuracy: 0.9052 - loss: 0.2817 - val_accuracy: 0.8254 - val_loss: 0.5106
Epoch 41/50
277/277             39s 81ms/step -
accuracy: 0.9071 - loss: 0.2791 - val_accuracy: 0.8241 - val_loss: 0.5189
Epoch 42/50
277/277             39s 75ms/step -
accuracy: 0.9100 - loss: 0.2690 - val_accuracy: 0.8245 - val_loss: 0.5270
Epoch 43/50
277/277             41s 74ms/step -
accuracy: 0.9153 - loss: 0.2649 - val_accuracy: 0.8272 - val_loss: 0.5369
Epoch 44/50
277/277             41s 75ms/step -
accuracy: 0.9171 - loss: 0.2640 - val_accuracy: 0.8254 - val_loss: 0.5494
Epoch 45/50
277/277             41s 76ms/step -
accuracy: 0.9192 - loss: 0.2583 - val_accuracy: 0.8248 - val_loss: 0.5621
Epoch 46/50
277/277             22s 80ms/step -
accuracy: 0.9248 - loss: 0.2481 - val_accuracy: 0.8257 - val_loss: 0.5703
Epoch 47/50
```

```
277/277             40s 78ms/step -
accuracy: 0.9259 - loss: 0.2420 - val_accuracy: 0.8245 - val_loss: 0.5752
Epoch 48/50
277/277             40s 75ms/step -
accuracy: 0.9315 - loss: 0.2394 - val_accuracy: 0.8236 - val_loss: 0.5903
Epoch 49/50
277/277             43s 81ms/step -
accuracy: 0.9321 - loss: 0.2327 - val_accuracy: 0.8245 - val_loss: 0.5962
Epoch 50/50
277/277             23s 84ms/step -
accuracy: 0.9355 - loss: 0.2269 - val_accuracy: 0.8261 - val_loss: 0.6126
139/139             1s 9ms/step
Fold 2 - Precision: 0.8703, Recall: 0.9051, F1 Score: 0.8873
Training for fold 3 on GossipCop…
Epoch 1/50
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
277/277             23s 74ms/step -
accuracy: 0.6654 - loss: 1.6911 - val_accuracy: 0.7545 - val_loss: 1.0713
Epoch 2/50
277/277             22s 80ms/step -
accuracy: 0.7603 - loss: 0.9795 - val_accuracy: 0.7545 - val_loss: 0.7577
Epoch 3/50
277/277             40s 78ms/step -
accuracy: 0.7603 - loss: 0.7262 - val_accuracy: 0.7545 - val_loss: 0.6378
Epoch 4/50
277/277             21s 76ms/step -
accuracy: 0.7603 - loss: 0.6296 - val_accuracy: 0.7545 - val_loss: 0.5931
Epoch 5/50
277/277             23s 81ms/step -
accuracy: 0.7603 - loss: 0.5911 - val_accuracy: 0.7545 - val_loss: 0.5763
Epoch 6/50
277/277             39s 75ms/step -
accuracy: 0.7603 - loss: 0.5769 - val_accuracy: 0.7545 - val_loss: 0.5694
Epoch 7/50
277/277             41s 74ms/step -
accuracy: 0.7603 - loss: 0.5744 - val_accuracy: 0.7545 - val_loss: 0.5664
Epoch 8/50
277/277             22s 80ms/step -
accuracy: 0.7603 - loss: 0.5694 - val_accuracy: 0.7545 - val_loss: 0.5645
Epoch 9/50
277/277             39s 75ms/step -
accuracy: 0.7603 - loss: 0.5674 - val_accuracy: 0.7545 - val_loss: 0.5632
Epoch 10/50
277/277             43s 81ms/step -
accuracy: 0.7603 - loss: 0.5668 - val_accuracy: 0.7545 - val_loss: 0.5622
```

```
Epoch 11/50
277/277            22s 79ms/step -
accuracy: 0.7603 - loss: 0.5646 - val_accuracy: 0.7545 - val_loss: 0.5615
Epoch 12/50
277/277            41s 81ms/step -
accuracy: 0.7603 - loss: 0.5647 - val_accuracy: 0.7545 - val_loss: 0.5610
Epoch 13/50
277/277            21s 74ms/step -
accuracy: 0.7603 - loss: 0.5638 - val_accuracy: 0.7545 - val_loss: 0.5608
Epoch 14/50
277/277            41s 74ms/step -
accuracy: 0.7603 - loss: 0.5634 - val_accuracy: 0.7545 - val_loss: 0.5604
Epoch 15/50
277/277            20s 74ms/step -
accuracy: 0.7603 - loss: 0.5619 - val_accuracy: 0.7545 - val_loss: 0.5601
Epoch 16/50
277/277            25s 90ms/step -
accuracy: 0.7603 - loss: 0.5622 - val_accuracy: 0.7545 - val_loss: 0.5602
Epoch 17/50
277/277            37s 74ms/step -
accuracy: 0.7603 - loss: 0.5621 - val_accuracy: 0.7545 - val_loss: 0.5598
Epoch 18/50
277/277            41s 75ms/step -
accuracy: 0.7603 - loss: 0.5586 - val_accuracy: 0.7545 - val_loss: 0.5596
Epoch 19/50
277/277            41s 75ms/step -
accuracy: 0.7603 - loss: 0.5605 - val_accuracy: 0.7545 - val_loss: 0.5594
Epoch 20/50
277/277            43s 80ms/step -
accuracy: 0.7603 - loss: 0.5587 - val_accuracy: 0.7545 - val_loss: 0.5592
Epoch 21/50
277/277            39s 75ms/step -
accuracy: 0.7603 - loss: 0.5608 - val_accuracy: 0.7545 - val_loss: 0.5591
Epoch 22/50
277/277            21s 76ms/step -
accuracy: 0.7603 - loss: 0.5608 - val_accuracy: 0.7545 - val_loss: 0.5591
Epoch 23/50
277/277            40s 74ms/step -
accuracy: 0.7603 - loss: 0.5596 - val_accuracy: 0.7545 - val_loss: 0.5589
Epoch 24/50
277/277            22s 80ms/step -
accuracy: 0.7603 - loss: 0.5580 - val_accuracy: 0.7545 - val_loss: 0.5588
Epoch 25/50
277/277            42s 86ms/step -
accuracy: 0.7603 - loss: 0.5597 - val_accuracy: 0.7545 - val_loss: 0.5589
Epoch 26/50
277/277            38s 75ms/step -
accuracy: 0.7603 - loss: 0.5583 - val_accuracy: 0.7545 - val_loss: 0.5587
```

```
Epoch 27/50
277/277            41s 74ms/step -
accuracy: 0.7603 - loss: 0.5600 - val_accuracy: 0.7545 - val_loss: 0.5588
Epoch 28/50
277/277            41s 74ms/step -
accuracy: 0.7603 - loss: 0.5591 - val_accuracy: 0.7545 - val_loss: 0.5586
Epoch 29/50
277/277            44s 87ms/step -
accuracy: 0.7603 - loss: 0.5576 - val_accuracy: 0.7545 - val_loss: 0.5585
Epoch 30/50
277/277            20s 73ms/step -
accuracy: 0.7603 - loss: 0.5579 - val_accuracy: 0.7545 - val_loss: 0.5585
Epoch 31/50
277/277            21s 73ms/step -
accuracy: 0.7603 - loss: 0.5591 - val_accuracy: 0.7545 - val_loss: 0.5585
Epoch 32/50
277/277            23s 81ms/step -
accuracy: 0.7603 - loss: 0.5577 - val_accuracy: 0.7545 - val_loss: 0.5583
Epoch 33/50
277/277            40s 77ms/step -
accuracy: 0.7603 - loss: 0.5575 - val_accuracy: 0.7545 - val_loss: 0.5585
Epoch 34/50
277/277            40s 75ms/step -
accuracy: 0.7603 - loss: 0.5579 - val_accuracy: 0.7545 - val_loss: 0.5584
Epoch 35/50
277/277            24s 87ms/step -
accuracy: 0.7603 - loss: 0.5576 - val_accuracy: 0.7545 - val_loss: 0.5582
Epoch 36/50
277/277            37s 74ms/step -
accuracy: 0.7603 - loss: 0.5569 - val_accuracy: 0.7545 - val_loss: 0.5586
Epoch 37/50
277/277            22s 81ms/step -
accuracy: 0.7603 - loss: 0.5583 - val_accuracy: 0.7545 - val_loss: 0.5582
Epoch 38/50
277/277            41s 82ms/step -
accuracy: 0.7603 - loss: 0.5581 - val_accuracy: 0.7545 - val_loss: 0.5581
Epoch 39/50
277/277            39s 75ms/step -
accuracy: 0.7603 - loss: 0.5566 - val_accuracy: 0.7545 - val_loss: 0.5582
Epoch 40/50
277/277            41s 74ms/step -
accuracy: 0.7603 - loss: 0.5574 - val_accuracy: 0.7545 - val_loss: 0.5582
Epoch 41/50
277/277            41s 74ms/step -
accuracy: 0.7603 - loss: 0.5569 - val_accuracy: 0.7545 - val_loss: 0.5580
Epoch 42/50
277/277            41s 74ms/step -
accuracy: 0.7603 - loss: 0.5570 - val_accuracy: 0.7545 - val_loss: 0.5581
```

```
Epoch 43/50
277/277             41s 74ms/step -
accuracy: 0.7603 - loss: 0.5569 - val_accuracy: 0.7545 - val_loss: 0.5580
Epoch 44/50
277/277             43s 81ms/step -
accuracy: 0.7603 - loss: 0.5565 - val_accuracy: 0.7545 - val_loss: 0.5580
Epoch 45/50
277/277             39s 75ms/step -
accuracy: 0.7603 - loss: 0.5549 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 46/50
277/277             41s 75ms/step -
accuracy: 0.7603 - loss: 0.5546 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 47/50
277/277             23s 82ms/step -
accuracy: 0.7603 - loss: 0.5573 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 48/50
277/277             40s 79ms/step -
accuracy: 0.7603 - loss: 0.5572 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 49/50
277/277             40s 74ms/step -
accuracy: 0.7603 - loss: 0.5540 - val_accuracy: 0.7545 - val_loss: 0.5578
Epoch 50/50
277/277             41s 74ms/step -
accuracy: 0.7603 - loss: 0.5572 - val_accuracy: 0.7545 - val_loss: 0.5578
139/139             1s 9ms/step
Fold 3 - Precision: 0.7545, Recall: 1.0000, F1 Score: 0.8601
Training for fold 4 on GossipCop…
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

277/277             25s 81ms/step -
accuracy: 0.6056 - loss: 1.7161 - val_accuracy: 0.7642 - val_loss: 1.0730
Epoch 2/50
277/277             20s 73ms/step -
accuracy: 0.7587 - loss: 0.9947 - val_accuracy: 0.7642 - val_loss: 0.7518
Epoch 3/50
277/277             21s 75ms/step -
accuracy: 0.7587 - loss: 0.7374 - val_accuracy: 0.7642 - val_loss: 0.6294
Epoch 4/50
277/277             40s 73ms/step -
accuracy: 0.7587 - loss: 0.6346 - val_accuracy: 0.7642 - val_loss: 0.5842
Epoch 5/50
277/277             22s 80ms/step -
accuracy: 0.7587 - loss: 0.5963 - val_accuracy: 0.7642 - val_loss: 0.5672
Epoch 6/50
277/277             40s 77ms/step -
```

```
accuracy: 0.7587 - loss: 0.5834 - val_accuracy: 0.7642 - val_loss: 0.5590
Epoch 7/50
277/277            42s 79ms/step -
accuracy: 0.7587 - loss: 0.5773 - val_accuracy: 0.7642 - val_loss: 0.5568
Epoch 8/50
277/277            40s 76ms/step -
accuracy: 0.7587 - loss: 0.5733 - val_accuracy: 0.7642 - val_loss: 0.5537
Epoch 9/50
277/277            23s 83ms/step -
accuracy: 0.7587 - loss: 0.5747 - val_accuracy: 0.7642 - val_loss: 0.5540
Epoch 10/50
277/277            39s 75ms/step -
accuracy: 0.7587 - loss: 0.5712 - val_accuracy: 0.7642 - val_loss: 0.5526
Epoch 11/50
277/277            41s 74ms/step -
accuracy: 0.7587 - loss: 0.5723 - val_accuracy: 0.7642 - val_loss: 0.5519
Epoch 12/50
277/277            41s 74ms/step -
accuracy: 0.7587 - loss: 0.5684 - val_accuracy: 0.7642 - val_loss: 0.5511
Epoch 13/50
277/277            23s 82ms/step -
accuracy: 0.7587 - loss: 0.5684 - val_accuracy: 0.7642 - val_loss: 0.5511
Epoch 14/50
277/277            39s 77ms/step -
accuracy: 0.7587 - loss: 0.5695 - val_accuracy: 0.7642 - val_loss: 0.5509
Epoch 15/50
277/277            40s 74ms/step -
accuracy: 0.7587 - loss: 0.5650 - val_accuracy: 0.7642 - val_loss: 0.5497
Epoch 16/50
277/277            22s 80ms/step -
accuracy: 0.7587 - loss: 0.5686 - val_accuracy: 0.7642 - val_loss: 0.5505
Epoch 17/50
277/277            40s 78ms/step -
accuracy: 0.7587 - loss: 0.5675 - val_accuracy: 0.7642 - val_loss: 0.5495
Epoch 18/50
277/277            42s 83ms/step -
accuracy: 0.7587 - loss: 0.5686 - val_accuracy: 0.7642 - val_loss: 0.5496
Epoch 19/50
277/277            39s 74ms/step -
accuracy: 0.7587 - loss: 0.5655 - val_accuracy: 0.7642 - val_loss: 0.5491
Epoch 20/50
277/277            41s 74ms/step -
accuracy: 0.7587 - loss: 0.5653 - val_accuracy: 0.7642 - val_loss: 0.5496
Epoch 21/50
277/277            41s 73ms/step -
accuracy: 0.7587 - loss: 0.5634 - val_accuracy: 0.7642 - val_loss: 0.5494
Epoch 22/50
277/277            23s 84ms/step -
```

```
accuracy: 0.7587 - loss: 0.5661 - val_accuracy: 0.7642 - val_loss: 0.5494
Epoch 23/50
277/277              41s 82ms/step -
accuracy: 0.7587 - loss: 0.5639 - val_accuracy: 0.7642 - val_loss: 0.5487
Epoch 24/50
277/277              22s 80ms/step -
accuracy: 0.7587 - loss: 0.5625 - val_accuracy: 0.7642 - val_loss: 0.5486
Epoch 25/50
277/277              42s 82ms/step -
accuracy: 0.7587 - loss: 0.5641 - val_accuracy: 0.7642 - val_loss: 0.5484
Epoch 26/50
277/277              39s 75ms/step -
accuracy: 0.7587 - loss: 0.5640 - val_accuracy: 0.7642 - val_loss: 0.5482
Epoch 27/50
277/277              41s 75ms/step -
accuracy: 0.7587 - loss: 0.5636 - val_accuracy: 0.7642 - val_loss: 0.5486
Epoch 28/50
277/277              43s 83ms/step -
accuracy: 0.7587 - loss: 0.5639 - val_accuracy: 0.7642 - val_loss: 0.5485
Epoch 29/50
277/277              39s 74ms/step -
accuracy: 0.7587 - loss: 0.5628 - val_accuracy: 0.7642 - val_loss: 0.5489
Epoch 30/50
277/277              41s 74ms/step -
accuracy: 0.7587 - loss: 0.5627 - val_accuracy: 0.7642 - val_loss: 0.5485
Epoch 31/50
277/277              41s 73ms/step -
accuracy: 0.7587 - loss: 0.5616 - val_accuracy: 0.7642 - val_loss: 0.5483
Epoch 32/50
277/277              24s 84ms/step -
accuracy: 0.7587 - loss: 0.5629 - val_accuracy: 0.7642 - val_loss: 0.5481
Epoch 33/50
277/277              39s 76ms/step -
accuracy: 0.7587 - loss: 0.5612 - val_accuracy: 0.7642 - val_loss: 0.5478
Epoch 34/50
277/277              41s 75ms/step -
accuracy: 0.7587 - loss: 0.5620 - val_accuracy: 0.7642 - val_loss: 0.5478
Epoch 35/50
277/277              22s 79ms/step -
accuracy: 0.7587 - loss: 0.5602 - val_accuracy: 0.7642 - val_loss: 0.5479
Epoch 36/50
277/277              42s 83ms/step -
accuracy: 0.7587 - loss: 0.5609 - val_accuracy: 0.7642 - val_loss: 0.5485
Epoch 37/50
277/277              39s 75ms/step -
accuracy: 0.7587 - loss: 0.5599 - val_accuracy: 0.7642 - val_loss: 0.5479
Epoch 38/50
277/277              41s 75ms/step -
```

```
accuracy: 0.7587 - loss: 0.5603 - val_accuracy: 0.7642 - val_loss: 0.5477
Epoch 39/50
277/277            22s 79ms/step -
accuracy: 0.7587 - loss: 0.5603 - val_accuracy: 0.7642 - val_loss: 0.5475
Epoch 40/50
277/277            20s 73ms/step -
accuracy: 0.7587 - loss: 0.5589 - val_accuracy: 0.7642 - val_loss: 0.5472
Epoch 41/50
277/277            22s 79ms/step -
accuracy: 0.7587 - loss: 0.5615 - val_accuracy: 0.7642 - val_loss: 0.5480
Epoch 42/50
277/277            40s 75ms/step -
accuracy: 0.7587 - loss: 0.5583 - val_accuracy: 0.7642 - val_loss: 0.5479
Epoch 43/50
277/277            41s 74ms/step -
accuracy: 0.7587 - loss: 0.5580 - val_accuracy: 0.7642 - val_loss: 0.5484
Epoch 44/50
277/277            24s 87ms/step -
accuracy: 0.7587 - loss: 0.5593 - val_accuracy: 0.7642 - val_loss: 0.5483
Epoch 45/50
277/277            37s 75ms/step -
accuracy: 0.7587 - loss: 0.5598 - val_accuracy: 0.7642 - val_loss: 0.5477
Epoch 46/50
277/277            41s 76ms/step -
accuracy: 0.7587 - loss: 0.5576 - val_accuracy: 0.7642 - val_loss: 0.5478
Epoch 47/50
277/277            40s 74ms/step -
accuracy: 0.7587 - loss: 0.5594 - val_accuracy: 0.7642 - val_loss: 0.5478
Epoch 48/50
277/277            44s 84ms/step -
accuracy: 0.7587 - loss: 0.5591 - val_accuracy: 0.7642 - val_loss: 0.5474
Epoch 49/50
277/277            22s 78ms/step -
accuracy: 0.7587 - loss: 0.5597 - val_accuracy: 0.7642 - val_loss: 0.5475
Epoch 50/50
277/277            41s 79ms/step -
accuracy: 0.7587 - loss: 0.5564 - val_accuracy: 0.7642 - val_loss: 0.5477
139/139            1s 9ms/step
Fold 4 - Precision: 0.7642, Recall: 1.0000, F1 Score: 0.8664
Training for fold 5 on GossipCop…
Epoch 1/50
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
277/277            25s 77ms/step -
accuracy: 0.7521 - loss: 1.6654 - val_accuracy: 0.7574 - val_loss: 1.0783
Epoch 2/50
```

```
277/277              21s 74ms/step -
accuracy: 0.7600 - loss: 0.9843 - val_accuracy: 0.7574 - val_loss: 0.7640
Epoch 3/50
277/277              41s 74ms/step -
accuracy: 0.7600 - loss: 0.7298 - val_accuracy: 0.7574 - val_loss: 0.6398
Epoch 4/50
277/277              25s 91ms/step -
accuracy: 0.7600 - loss: 0.6332 - val_accuracy: 0.7574 - val_loss: 0.5928
Epoch 5/50
277/277              36s 73ms/step -
accuracy: 0.7600 - loss: 0.5925 - val_accuracy: 0.7574 - val_loss: 0.5743
Epoch 6/50
277/277              23s 83ms/step -
accuracy: 0.7600 - loss: 0.5766 - val_accuracy: 0.7574 - val_loss: 0.5626
Epoch 7/50
277/277              39s 76ms/step -
accuracy: 0.7612 - loss: 0.5553 - val_accuracy: 0.7990 - val_loss: 0.4870
Epoch 8/50
277/277              41s 74ms/step -
accuracy: 0.8111 - loss: 0.4608 - val_accuracy: 0.8240 - val_loss: 0.4558
Epoch 9/50
277/277              43s 81ms/step -
accuracy: 0.8420 - loss: 0.4137 - val_accuracy: 0.8301 - val_loss: 0.4451
Epoch 10/50
277/277              39s 75ms/step -
accuracy: 0.8617 - loss: 0.3873 - val_accuracy: 0.8306 - val_loss: 0.4416
Epoch 11/50
277/277              40s 73ms/step -
accuracy: 0.8726 - loss: 0.3649 - val_accuracy: 0.8308 - val_loss: 0.4407
Epoch 12/50
277/277              23s 83ms/step -
accuracy: 0.8819 - loss: 0.3469 - val_accuracy: 0.8335 - val_loss: 0.4441
Epoch 13/50
277/277              39s 75ms/step -
accuracy: 0.8858 - loss: 0.3388 - val_accuracy: 0.8333 - val_loss: 0.4484
Epoch 14/50
277/277              24s 88ms/step -
accuracy: 0.8935 - loss: 0.3237 - val_accuracy: 0.8326 - val_loss: 0.4516
Epoch 15/50
277/277              20s 73ms/step -
accuracy: 0.8968 - loss: 0.3165 - val_accuracy: 0.8331 - val_loss: 0.4565
Epoch 16/50
277/277              21s 76ms/step -
accuracy: 0.9011 - loss: 0.3065 - val_accuracy: 0.8342 - val_loss: 0.4605
Epoch 17/50
277/277              22s 78ms/step -
accuracy: 0.9056 - loss: 0.2997 - val_accuracy: 0.8306 - val_loss: 0.4676
Epoch 18/50
```

```
277/277             42s 80ms/step -
accuracy: 0.9061 - loss: 0.2904 - val_accuracy: 0.8313 - val_loss: 0.4735
Epoch 19/50
277/277             40s 75ms/step -
accuracy: 0.9107 - loss: 0.2856 - val_accuracy: 0.8292 - val_loss: 0.4821
Epoch 20/50
277/277             43s 84ms/step -
accuracy: 0.9138 - loss: 0.2779 - val_accuracy: 0.8292 - val_loss: 0.4897
Epoch 21/50
277/277             38s 74ms/step -
accuracy: 0.9144 - loss: 0.2747 - val_accuracy: 0.8295 - val_loss: 0.4983
Epoch 22/50
277/277             23s 84ms/step -
accuracy: 0.9191 - loss: 0.2687 - val_accuracy: 0.8279 - val_loss: 0.5039
Epoch 23/50
277/277             20s 74ms/step -
accuracy: 0.9235 - loss: 0.2609 - val_accuracy: 0.8274 - val_loss: 0.5135
Epoch 24/50
277/277             21s 75ms/step -
accuracy: 0.9247 - loss: 0.2556 - val_accuracy: 0.8240 - val_loss: 0.5172
Epoch 25/50
277/277             41s 75ms/step -
accuracy: 0.9271 - loss: 0.2524 - val_accuracy: 0.8254 - val_loss: 0.5246
Epoch 26/50
277/277             43s 83ms/step -
accuracy: 0.9283 - loss: 0.2463 - val_accuracy: 0.8215 - val_loss: 0.5353
Epoch 27/50
277/277             39s 75ms/step -
accuracy: 0.9301 - loss: 0.2439 - val_accuracy: 0.8227 - val_loss: 0.5398
Epoch 28/50
277/277             41s 74ms/step -
accuracy: 0.9316 - loss: 0.2375 - val_accuracy: 0.8234 - val_loss: 0.5515
Epoch 29/50
277/277             21s 75ms/step -
accuracy: 0.9352 - loss: 0.2306 - val_accuracy: 0.8222 - val_loss: 0.5577
Epoch 30/50
277/277             41s 75ms/step -
accuracy: 0.9373 - loss: 0.2250 - val_accuracy: 0.8227 - val_loss: 0.5723
Epoch 31/50
277/277             42s 80ms/step -
accuracy: 0.9394 - loss: 0.2196 - val_accuracy: 0.8225 - val_loss: 0.5828
Epoch 32/50
277/277             39s 75ms/step -
accuracy: 0.9446 - loss: 0.2169 - val_accuracy: 0.8225 - val_loss: 0.5937
Epoch 33/50
277/277             41s 75ms/step -
accuracy: 0.9439 - loss: 0.2116 - val_accuracy: 0.8211 - val_loss: 0.5984
Epoch 34/50
```

```
277/277             41s 75ms/step -
accuracy: 0.9481 - loss: 0.2057 - val_accuracy: 0.8197 - val_loss: 0.6167
Epoch 35/50
277/277             41s 74ms/step -
accuracy: 0.9488 - loss: 0.2012 - val_accuracy: 0.8193 - val_loss: 0.6277
Epoch 36/50
277/277             41s 75ms/step -
accuracy: 0.9527 - loss: 0.1956 - val_accuracy: 0.8179 - val_loss: 0.6435
Epoch 37/50
277/277             41s 74ms/step -
accuracy: 0.9564 - loss: 0.1889 - val_accuracy: 0.8173 - val_loss: 0.6620
Epoch 38/50
277/277             41s 74ms/step -
accuracy: 0.9563 - loss: 0.1848 - val_accuracy: 0.8175 - val_loss: 0.6725
Epoch 39/50
277/277             41s 74ms/step -
accuracy: 0.9592 - loss: 0.1831 - val_accuracy: 0.8161 - val_loss: 0.6842
Epoch 40/50
277/277             43s 82ms/step -
accuracy: 0.9610 - loss: 0.1767 - val_accuracy: 0.8164 - val_loss: 0.6948
Epoch 41/50
277/277             22s 81ms/step -
accuracy: 0.9631 - loss: 0.1732 - val_accuracy: 0.8161 - val_loss: 0.7112
Epoch 42/50
277/277             40s 76ms/step -
accuracy: 0.9636 - loss: 0.1693 - val_accuracy: 0.8159 - val_loss: 0.7305
Epoch 43/50
277/277             22s 79ms/step -
accuracy: 0.9665 - loss: 0.1635 - val_accuracy: 0.8157 - val_loss: 0.7394
Epoch 44/50
277/277             41s 81ms/step -
accuracy: 0.9692 - loss: 0.1593 - val_accuracy: 0.8157 - val_loss: 0.7501
Epoch 45/50
277/277             42s 84ms/step -
accuracy: 0.9691 - loss: 0.1557 - val_accuracy: 0.8139 - val_loss: 0.7634
Epoch 46/50
277/277             22s 80ms/step -
accuracy: 0.9693 - loss: 0.1520 - val_accuracy: 0.8139 - val_loss: 0.7789
Epoch 47/50
277/277             41s 79ms/step -
accuracy: 0.9724 - loss: 0.1476 - val_accuracy: 0.8166 - val_loss: 0.8010
Epoch 48/50
277/277             41s 80ms/step -
accuracy: 0.9718 - loss: 0.1435 - val_accuracy: 0.8173 - val_loss: 0.8234
Epoch 49/50
277/277             39s 74ms/step -
accuracy: 0.9736 - loss: 0.1396 - val_accuracy: 0.8150 - val_loss: 0.8232
Epoch 50/50
```

```
277/277             22s 81ms/step -
accuracy: 0.9732 - loss: 0.1386 - val_accuracy: 0.8116 - val_loss: 0.8188
139/139             1s 10ms/step
Fold 5 - Precision: 0.8675, Recall: 0.8867, F1 Score: 0.8770
Average Precision: 0.8242
Average Recall: 0.9412
Average F1 Score: 0.8759
Training for fold 1 on PolitiFact…
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

14/14             6s 139ms/step -
accuracy: 0.5793 - loss: 3.4624 - val_accuracy: 0.6161 - val_loss: 3.3594
Epoch 2/50
14/14             2s 116ms/step -
accuracy: 0.5982 - loss: 3.3315 - val_accuracy: 0.6161 - val_loss: 3.2334
Epoch 3/50
14/14             2s 112ms/step -
accuracy: 0.5910 - loss: 3.2054 - val_accuracy: 0.6161 - val_loss: 3.1132
Epoch 4/50
14/14             2s 112ms/step -
accuracy: 0.5936 - loss: 3.0865 - val_accuracy: 0.6161 - val_loss: 2.9989
Epoch 5/50
14/14             2s 117ms/step -
accuracy: 0.5896 - loss: 2.9759 - val_accuracy: 0.6161 - val_loss: 2.8906
Epoch 6/50
14/14             2s 171ms/step -
accuracy: 0.5880 - loss: 2.8654 - val_accuracy: 0.6161 - val_loss: 2.7875
Epoch 7/50
14/14             3s 184ms/step -
accuracy: 0.5869 - loss: 2.7640 - val_accuracy: 0.6161 - val_loss: 2.6897
Epoch 8/50
14/14             4s 120ms/step -
accuracy: 0.5874 - loss: 2.6670 - val_accuracy: 0.6161 - val_loss: 2.5964
Epoch 9/50
14/14             2s 116ms/step -
accuracy: 0.5874 - loss: 2.5766 - val_accuracy: 0.6161 - val_loss: 2.5080
Epoch 10/50
14/14             2s 112ms/step -
accuracy: 0.5904 - loss: 2.4879 - val_accuracy: 0.6161 - val_loss: 2.4241
Epoch 11/50
14/14             3s 175ms/step -
accuracy: 0.5882 - loss: 2.4035 - val_accuracy: 0.6161 - val_loss: 2.3440
Epoch 12/50
14/14             2s 175ms/step -
accuracy: 0.5894 - loss: 2.3246 - val_accuracy: 0.6161 - val_loss: 2.2676
```

```
Epoch 13/50
14/14            2s 113ms/step -
accuracy: 0.5892 - loss: 2.2535 - val_accuracy: 0.6161 - val_loss: 2.1956
Epoch 14/50
14/14            2s 115ms/step -
accuracy: 0.5936 - loss: 2.1781 - val_accuracy: 0.6161 - val_loss: 2.1268
Epoch 15/50
14/14            3s 117ms/step -
accuracy: 0.5924 - loss: 2.1102 - val_accuracy: 0.6161 - val_loss: 2.0611
Epoch 16/50
14/14            2s 117ms/step -
accuracy: 0.5931 - loss: 2.0443 - val_accuracy: 0.6161 - val_loss: 1.9985
Epoch 17/50
14/14            3s 120ms/step -
accuracy: 0.5903 - loss: 1.9839 - val_accuracy: 0.6161 - val_loss: 1.9391
Epoch 18/50
14/14            3s 170ms/step -
accuracy: 0.5906 - loss: 1.9259 - val_accuracy: 0.6161 - val_loss: 1.8826
Epoch 19/50
14/14            2s 138ms/step -
accuracy: 0.5909 - loss: 1.8688 - val_accuracy: 0.6161 - val_loss: 1.8285
Epoch 20/50
14/14            2s 114ms/step -
accuracy: 0.5922 - loss: 1.8115 - val_accuracy: 0.6161 - val_loss: 1.7771
Epoch 21/50
14/14            2s 117ms/step -
accuracy: 0.5918 - loss: 1.7648 - val_accuracy: 0.6161 - val_loss: 1.7280
Epoch 22/50
14/14            2s 119ms/step -
accuracy: 0.5971 - loss: 1.7133 - val_accuracy: 0.6161 - val_loss: 1.6810
Epoch 23/50
14/14            2s 113ms/step -
accuracy: 0.6190 - loss: 1.6669 - val_accuracy: 0.6161 - val_loss: 1.6368
Epoch 24/50
14/14            2s 129ms/step -
accuracy: 0.6374 - loss: 1.6226 - val_accuracy: 0.6161 - val_loss: 1.5944
Epoch 25/50
14/14            3s 185ms/step -
accuracy: 0.6555 - loss: 1.5774 - val_accuracy: 0.6161 - val_loss: 1.5528
Epoch 26/50
14/14            4s 118ms/step -
accuracy: 0.6578 - loss: 1.5365 - val_accuracy: 0.6161 - val_loss: 1.5130
Epoch 27/50
14/14            3s 116ms/step -
accuracy: 0.6838 - loss: 1.4930 - val_accuracy: 0.6493 - val_loss: 1.4747
Epoch 28/50
14/14            2s 112ms/step -
accuracy: 0.7153 - loss: 1.4526 - val_accuracy: 0.6682 - val_loss: 1.4375
```

```
Epoch 29/50
14/14            3s 148ms/step -
accuracy: 0.7424 - loss: 1.4145 - val_accuracy: 0.6872 - val_loss: 1.4004
Epoch 30/50
14/14            3s 175ms/step -
accuracy: 0.7691 - loss: 1.3732 - val_accuracy: 0.7251 - val_loss: 1.3637
Epoch 31/50
14/14            2s 124ms/step -
accuracy: 0.7980 - loss: 1.3360 - val_accuracy: 0.7488 - val_loss: 1.3276
Epoch 32/50
14/14            2s 117ms/step -
accuracy: 0.8040 - loss: 1.2906 - val_accuracy: 0.7536 - val_loss: 1.2915
Epoch 33/50
14/14            3s 115ms/step -
accuracy: 0.8465 - loss: 1.2452 - val_accuracy: 0.7678 - val_loss: 1.2556
Epoch 34/50
14/14            2s 117ms/step -
accuracy: 0.8608 - loss: 1.2008 - val_accuracy: 0.7773 - val_loss: 1.2198
Epoch 35/50
14/14            3s 143ms/step -
accuracy: 0.8880 - loss: 1.1477 - val_accuracy: 0.7962 - val_loss: 1.1844
Epoch 36/50
14/14            3s 181ms/step -
accuracy: 0.8939 - loss: 1.0910 - val_accuracy: 0.7962 - val_loss: 1.1501
Epoch 37/50
14/14            4s 117ms/step -
accuracy: 0.8984 - loss: 1.0541 - val_accuracy: 0.8104 - val_loss: 1.1183
Epoch 38/50
14/14            2s 113ms/step -
accuracy: 0.9046 - loss: 1.0022 - val_accuracy: 0.8152 - val_loss: 1.0908
Epoch 39/50
14/14            2s 117ms/step -
accuracy: 0.9266 - loss: 0.9479 - val_accuracy: 0.8199 - val_loss: 1.0656
Epoch 40/50
14/14            3s 135ms/step -
accuracy: 0.9411 - loss: 0.9188 - val_accuracy: 0.8199 - val_loss: 1.0435
Epoch 41/50
14/14            3s 185ms/step -
accuracy: 0.9499 - loss: 0.8638 - val_accuracy: 0.8341 - val_loss: 1.0215
Epoch 42/50
14/14            4s 112ms/step -
accuracy: 0.9476 - loss: 0.8538 - val_accuracy: 0.8246 - val_loss: 1.0043
Epoch 43/50
14/14            2s 113ms/step -
accuracy: 0.9549 - loss: 0.8111 - val_accuracy: 0.8246 - val_loss: 0.9870
Epoch 44/50
14/14            3s 121ms/step -
accuracy: 0.9600 - loss: 0.7987 - val_accuracy: 0.8294 - val_loss: 0.9715
```

```
Epoch 45/50
14/14            2s 117ms/step -
accuracy: 0.9647 - loss: 0.7633 - val_accuracy: 0.8341 - val_loss: 0.9556
Epoch 46/50
14/14            2s 151ms/step -
accuracy: 0.9636 - loss: 0.7371 - val_accuracy: 0.8389 - val_loss: 0.9350
Epoch 47/50
14/14            3s 176ms/step -
accuracy: 0.9664 - loss: 0.7146 - val_accuracy: 0.8483 - val_loss: 0.9256
Epoch 48/50
14/14            2s 116ms/step -
accuracy: 0.9691 - loss: 0.6973 - val_accuracy: 0.8436 - val_loss: 0.9109
Epoch 49/50
14/14            2s 117ms/step -
accuracy: 0.9635 - loss: 0.6743 - val_accuracy: 0.8483 - val_loss: 0.8954
Epoch 50/50
14/14            2s 117ms/step -
accuracy: 0.9720 - loss: 0.6500 - val_accuracy: 0.8578 - val_loss: 0.8830
7/7            0s 30ms/step
Fold 1 - Precision: 0.8472, Recall: 0.9385, F1 Score: 0.8905
Training for fold 2 on PolitiFact…
Epoch 1/50
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
14/14            5s 157ms/step -
accuracy: 0.5861 - loss: 3.4623 - val_accuracy: 0.6066 - val_loss: 3.3619
Epoch 2/50
14/14            3s 183ms/step -
accuracy: 0.5967 - loss: 3.3321 - val_accuracy: 0.6066 - val_loss: 3.2362
Epoch 3/50
14/14            2s 141ms/step -
accuracy: 0.5877 - loss: 3.2075 - val_accuracy: 0.6066 - val_loss: 3.1163
Epoch 4/50
14/14            2s 111ms/step -
accuracy: 0.5822 - loss: 3.0897 - val_accuracy: 0.6066 - val_loss: 3.0020
Epoch 5/50
14/14            2s 112ms/step -
accuracy: 0.5840 - loss: 2.9763 - val_accuracy: 0.6066 - val_loss: 2.8932
Epoch 6/50
14/14            2s 111ms/step -
accuracy: 0.5892 - loss: 2.8688 - val_accuracy: 0.6066 - val_loss: 2.7898
Epoch 7/50
14/14            3s 113ms/step -
accuracy: 0.5844 - loss: 2.7663 - val_accuracy: 0.6066 - val_loss: 2.6915
Epoch 8/50
14/14            2s 114ms/step -
```

```
accuracy: 0.5841 - loss: 2.6685 - val_accuracy: 0.6066 - val_loss: 2.5978
Epoch 9/50
14/14            2s 167ms/step -
accuracy: 0.5839 - loss: 2.5763 - val_accuracy: 0.6066 - val_loss: 2.5089
Epoch 10/50
14/14            3s 182ms/step -
accuracy: 0.5833 - loss: 2.4913 - val_accuracy: 0.6066 - val_loss: 2.4247
Epoch 11/50
14/14            4s 115ms/step -
accuracy: 0.5829 - loss: 2.4094 - val_accuracy: 0.6066 - val_loss: 2.3448
Epoch 12/50
14/14            2s 113ms/step -
accuracy: 0.5833 - loss: 2.3281 - val_accuracy: 0.6066 - val_loss: 2.2685
Epoch 13/50
14/14            2s 113ms/step -
accuracy: 0.5830 - loss: 2.2545 - val_accuracy: 0.6066 - val_loss: 2.1961
Epoch 14/50
14/14            2s 117ms/step -
accuracy: 0.5863 - loss: 2.1807 - val_accuracy: 0.6066 - val_loss: 2.1272
Epoch 15/50
14/14            3s 167ms/step -
accuracy: 0.5835 - loss: 2.1132 - val_accuracy: 0.6066 - val_loss: 2.0618
Epoch 16/50
14/14            3s 186ms/step -
accuracy: 0.5855 - loss: 2.0467 - val_accuracy: 0.6066 - val_loss: 1.9994
Epoch 17/50
14/14            4s 115ms/step -
accuracy: 0.5840 - loss: 1.9884 - val_accuracy: 0.6066 - val_loss: 1.9401
Epoch 18/50
14/14            2s 111ms/step -
accuracy: 0.5848 - loss: 1.9275 - val_accuracy: 0.6066 - val_loss: 1.8839
Epoch 19/50
14/14            2s 117ms/step -
accuracy: 0.5830 - loss: 1.8733 - val_accuracy: 0.6066 - val_loss: 1.8302
Epoch 20/50
14/14            2s 111ms/step -
accuracy: 0.5852 - loss: 1.8186 - val_accuracy: 0.6066 - val_loss: 1.7792
Epoch 21/50
14/14            3s 169ms/step -
accuracy: 0.5861 - loss: 1.7723 - val_accuracy: 0.6066 - val_loss: 1.7309
Epoch 22/50
14/14            3s 199ms/step -
accuracy: 0.5878 - loss: 1.7228 - val_accuracy: 0.6066 - val_loss: 1.6848
Epoch 23/50
14/14            4s 115ms/step -
accuracy: 0.5898 - loss: 1.6755 - val_accuracy: 0.6066 - val_loss: 1.6407
Epoch 24/50
14/14            3s 113ms/step -
```

```
accuracy: 0.5894 - loss: 1.6321 - val_accuracy: 0.6066 - val_loss: 1.5986
Epoch 25/50
14/14            3s 116ms/step -
accuracy: 0.5875 - loss: 1.5901 - val_accuracy: 0.6066 - val_loss: 1.5586
Epoch 26/50
14/14            2s 111ms/step -
accuracy: 0.5879 - loss: 1.5541 - val_accuracy: 0.6066 - val_loss: 1.5206
Epoch 27/50
14/14            2s 175ms/step -
accuracy: 0.5860 - loss: 1.5160 - val_accuracy: 0.6066 - val_loss: 1.4844
Epoch 28/50
14/14            3s 184ms/step -
accuracy: 0.5855 - loss: 1.4758 - val_accuracy: 0.6066 - val_loss: 1.4497
Epoch 29/50
14/14            4s 114ms/step -
accuracy: 0.5890 - loss: 1.4457 - val_accuracy: 0.6066 - val_loss: 1.4167
Epoch 30/50
14/14            3s 121ms/step -
accuracy: 0.5940 - loss: 1.4129 - val_accuracy: 0.6066 - val_loss: 1.3853
Epoch 31/50
14/14            2s 115ms/step -
accuracy: 0.6031 - loss: 1.3820 - val_accuracy: 0.6066 - val_loss: 1.3549
Epoch 32/50
14/14            3s 171ms/step -
accuracy: 0.5984 - loss: 1.3513 - val_accuracy: 0.6066 - val_loss: 1.3258
Epoch 33/50
14/14            3s 173ms/step -
accuracy: 0.6105 - loss: 1.3154 - val_accuracy: 0.6066 - val_loss: 1.2972
Epoch 34/50
14/14            2s 118ms/step -
accuracy: 0.6243 - loss: 1.2926 - val_accuracy: 0.6066 - val_loss: 1.2698
Epoch 35/50
14/14            2s 115ms/step -
accuracy: 0.6533 - loss: 1.2595 - val_accuracy: 0.6114 - val_loss: 1.2427
Epoch 36/50
14/14            2s 113ms/step -
accuracy: 0.6815 - loss: 1.2342 - val_accuracy: 0.6445 - val_loss: 1.2159
Epoch 37/50
14/14            2s 118ms/step -
accuracy: 0.7274 - loss: 1.1986 - val_accuracy: 0.7441 - val_loss: 1.1882
Epoch 38/50
14/14            2s 114ms/step -
accuracy: 0.7520 - loss: 1.1724 - val_accuracy: 0.7867 - val_loss: 1.1592
Epoch 39/50
14/14            2s 110ms/step -
accuracy: 0.8034 - loss: 1.1306 - val_accuracy: 0.8294 - val_loss: 1.1282
Epoch 40/50
14/14            3s 172ms/step -
```

```
accuracy: 0.8286 - loss: 1.0948 - val_accuracy: 0.8436 - val_loss: 1.0932
Epoch 41/50
14/14            2s 170ms/step -
accuracy: 0.8605 - loss: 1.0548 - val_accuracy: 0.8578 - val_loss: 1.0562
Epoch 42/50
14/14            2s 113ms/step -
accuracy: 0.8864 - loss: 0.9946 - val_accuracy: 0.8673 - val_loss: 1.0169
Epoch 43/50
14/14            3s 115ms/step -
accuracy: 0.9043 - loss: 0.9448 - val_accuracy: 0.8768 - val_loss: 0.9771
Epoch 44/50
14/14            3s 120ms/step -
accuracy: 0.9278 - loss: 0.8802 - val_accuracy: 0.8626 - val_loss: 0.9395
Epoch 45/50
14/14            2s 114ms/step -
accuracy: 0.9391 - loss: 0.8303 - val_accuracy: 0.8720 - val_loss: 0.9049
Epoch 46/50
14/14            2s 115ms/step -
accuracy: 0.9398 - loss: 0.7910 - val_accuracy: 0.8720 - val_loss: 0.8745
Epoch 47/50
14/14            2s 172ms/step -
accuracy: 0.9517 - loss: 0.7573 - val_accuracy: 0.8768 - val_loss: 0.8514
Epoch 48/50
14/14            3s 185ms/step -
accuracy: 0.9589 - loss: 0.7222 - val_accuracy: 0.8626 - val_loss: 0.8303
Epoch 49/50
14/14            2s 117ms/step -
accuracy: 0.9468 - loss: 0.6982 - val_accuracy: 0.8720 - val_loss: 0.8147
Epoch 50/50
14/14            3s 117ms/step -
accuracy: 0.9640 - loss: 0.6687 - val_accuracy: 0.8720 - val_loss: 0.8020

WARNING:tensorflow:5 out of the last 147 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x792c7929de10> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

7/7            0s 33ms/step
Fold 2 - Precision: 0.8976, Recall: 0.8906, F1 Score: 0.8941
Training for fold 3 on PolitiFact…
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
```

```
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
14/14            5s 149ms/step -
accuracy: 0.5788 - loss: 3.4714 - val_accuracy: 0.5640 - val_loss: 3.3727
Epoch 2/50
14/14            3s 184ms/step -
accuracy: 0.6267 - loss: 3.3393 - val_accuracy: 0.5640 - val_loss: 3.2472
Epoch 3/50
14/14            4s 114ms/step -
accuracy: 0.6010 - loss: 3.2146 - val_accuracy: 0.5640 - val_loss: 3.1274
Epoch 4/50
14/14            3s 116ms/step -
accuracy: 0.6072 - loss: 3.0934 - val_accuracy: 0.5640 - val_loss: 3.0132
Epoch 5/50
14/14            2s 115ms/step -
accuracy: 0.6119 - loss: 2.9814 - val_accuracy: 0.5640 - val_loss: 2.9048
Epoch 6/50
14/14            2s 112ms/step -
accuracy: 0.6073 - loss: 2.8725 - val_accuracy: 0.5640 - val_loss: 2.8017
Epoch 7/50
14/14            2s 169ms/step -
accuracy: 0.6055 - loss: 2.7701 - val_accuracy: 0.5640 - val_loss: 2.7038
Epoch 8/50
14/14            3s 185ms/step -
accuracy: 0.6060 - loss: 2.6749 - val_accuracy: 0.5640 - val_loss: 2.6106
Epoch 9/50
14/14            4s 116ms/step -
accuracy: 0.6001 - loss: 2.5817 - val_accuracy: 0.5640 - val_loss: 2.5222
Epoch 10/50
14/14            3s 114ms/step -
accuracy: 0.6092 - loss: 2.4949 - val_accuracy: 0.5640 - val_loss: 2.4381
Epoch 11/50
14/14            2s 118ms/step -
accuracy: 0.6044 - loss: 2.4083 - val_accuracy: 0.5640 - val_loss: 2.3584
Epoch 12/50
14/14            3s 134ms/step -
accuracy: 0.6072 - loss: 2.3301 - val_accuracy: 0.5640 - val_loss: 2.2823
Epoch 13/50
14/14            3s 188ms/step -
accuracy: 0.6065 - loss: 2.2516 - val_accuracy: 0.5640 - val_loss: 2.2102
Epoch 14/50
14/14            4s 116ms/step -
accuracy: 0.6012 - loss: 2.1827 - val_accuracy: 0.5640 - val_loss: 2.1415
Epoch 15/50
14/14            2s 114ms/step -
accuracy: 0.6061 - loss: 2.1123 - val_accuracy: 0.5640 - val_loss: 2.0763
Epoch 16/50
```

```
14/14            3s 116ms/step -
accuracy: 0.6085 - loss: 2.0466 - val_accuracy: 0.5640 - val_loss: 2.0146
Epoch 17/50
14/14            2s 116ms/step -
accuracy: 0.6056 - loss: 1.9851 - val_accuracy: 0.5640 - val_loss: 1.9560
Epoch 18/50
14/14            2s 170ms/step -
accuracy: 0.6042 - loss: 1.9287 - val_accuracy: 0.5640 - val_loss: 1.8996
Epoch 19/50
14/14            3s 183ms/step -
accuracy: 0.6064 - loss: 1.8712 - val_accuracy: 0.5640 - val_loss: 1.8459
Epoch 20/50
14/14            2s 121ms/step -
accuracy: 0.6050 - loss: 1.8213 - val_accuracy: 0.5640 - val_loss: 1.7949
Epoch 21/50
14/14            2s 113ms/step -
accuracy: 0.6095 - loss: 1.7673 - val_accuracy: 0.5640 - val_loss: 1.7463
Epoch 22/50
14/14            3s 116ms/step -
accuracy: 0.6042 - loss: 1.7152 - val_accuracy: 0.5640 - val_loss: 1.6999
Epoch 23/50
14/14            3s 117ms/step -
accuracy: 0.6145 - loss: 1.6759 - val_accuracy: 0.5640 - val_loss: 1.6555
Epoch 24/50
14/14            2s 137ms/step -
accuracy: 0.6351 - loss: 1.6228 - val_accuracy: 0.5640 - val_loss: 1.6133
Epoch 25/50
14/14            3s 182ms/step -
accuracy: 0.6358 - loss: 1.5857 - val_accuracy: 0.5640 - val_loss: 1.5731
Epoch 26/50
14/14            2s 157ms/step -
accuracy: 0.6319 - loss: 1.5464 - val_accuracy: 0.5640 - val_loss: 1.5340
Epoch 27/50
14/14            2s 113ms/step -
accuracy: 0.6569 - loss: 1.5016 - val_accuracy: 0.5640 - val_loss: 1.4964
Epoch 28/50
14/14            2s 113ms/step -
accuracy: 0.6878 - loss: 1.4626 - val_accuracy: 0.5735 - val_loss: 1.4601
Epoch 29/50
14/14            2s 116ms/step -
accuracy: 0.7247 - loss: 1.4194 - val_accuracy: 0.6351 - val_loss: 1.4246
Epoch 30/50
14/14            2s 118ms/step -
accuracy: 0.7363 - loss: 1.3826 - val_accuracy: 0.7014 - val_loss: 1.3900
Epoch 31/50
14/14            2s 112ms/step -
accuracy: 0.7433 - loss: 1.3402 - val_accuracy: 0.7393 - val_loss: 1.3554
Epoch 32/50
```

```
14/14            3s 171ms/step -
accuracy: 0.8000 - loss: 1.2934 - val_accuracy: 0.7536 - val_loss: 1.3211
Epoch 33/50
14/14            3s 174ms/step -
accuracy: 0.7977 - loss: 1.2500 - val_accuracy: 0.7583 - val_loss: 1.2870
Epoch 34/50
14/14            2s 112ms/step -
accuracy: 0.8516 - loss: 1.1916 - val_accuracy: 0.7773 - val_loss: 1.2528
Epoch 35/50
14/14            2s 113ms/step -
accuracy: 0.8617 - loss: 1.1455 - val_accuracy: 0.7630 - val_loss: 1.2196
Epoch 36/50
14/14            3s 116ms/step -
accuracy: 0.8969 - loss: 1.0924 - val_accuracy: 0.7867 - val_loss: 1.1881
Epoch 37/50
14/14            3s 114ms/step -
accuracy: 0.9081 - loss: 1.0433 - val_accuracy: 0.7962 - val_loss: 1.1596
Epoch 38/50
14/14            3s 158ms/step -
accuracy: 0.9283 - loss: 0.9854 - val_accuracy: 0.8009 - val_loss: 1.1346
Epoch 39/50
14/14            3s 185ms/step -
accuracy: 0.9295 - loss: 0.9427 - val_accuracy: 0.8009 - val_loss: 1.1121
Epoch 40/50
14/14            4s 112ms/step -
accuracy: 0.9483 - loss: 0.9036 - val_accuracy: 0.8057 - val_loss: 1.0926
Epoch 41/50
14/14            2s 112ms/step -
accuracy: 0.9438 - loss: 0.8696 - val_accuracy: 0.8057 - val_loss: 1.0753
Epoch 42/50
14/14            3s 118ms/step -
accuracy: 0.9518 - loss: 0.8452 - val_accuracy: 0.8152 - val_loss: 1.0584
Epoch 43/50
14/14            2s 113ms/step -
accuracy: 0.9624 - loss: 0.7980 - val_accuracy: 0.8009 - val_loss: 1.0449
Epoch 44/50
14/14            2s 182ms/step -
accuracy: 0.9632 - loss: 0.7851 - val_accuracy: 0.8104 - val_loss: 1.0296
Epoch 45/50
14/14            2s 175ms/step -
accuracy: 0.9616 - loss: 0.7684 - val_accuracy: 0.8152 - val_loss: 1.0131
Epoch 46/50
14/14            2s 122ms/step -
accuracy: 0.9587 - loss: 0.7375 - val_accuracy: 0.8057 - val_loss: 1.0000
Epoch 47/50
14/14            3s 142ms/step -
accuracy: 0.9697 - loss: 0.7110 - val_accuracy: 0.8104 - val_loss: 0.9861
Epoch 48/50
```

```
14/14            3s 186ms/step -
accuracy: 0.9663 - loss: 0.7014 - val_accuracy: 0.8152 - val_loss: 0.9707
Epoch 49/50
14/14            2s 120ms/step -
accuracy: 0.9676 - loss: 0.6806 - val_accuracy: 0.8199 - val_loss: 0.9531
Epoch 50/50
14/14            2s 135ms/step -
accuracy: 0.9674 - loss: 0.6791 - val_accuracy: 0.8199 - val_loss: 0.9374
```

```
WARNING:tensorflow:5 out of the last 15 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x792c7930e050> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
```

```
7/7            0s 44ms/step
Fold 3 - Precision: 0.8293, Recall: 0.8571, F1 Score: 0.8430
Training for fold 4 on PolitiFact…
Epoch 1/50
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
14/14            5s 132ms/step -
accuracy: 0.5130 - loss: 3.4579 - val_accuracy: 0.5687 - val_loss: 3.3594
Epoch 2/50
14/14            2s 115ms/step -
accuracy: 0.5385 - loss: 3.3281 - val_accuracy: 0.5782 - val_loss: 3.2340
Epoch 3/50
14/14            2s 114ms/step -
accuracy: 0.5679 - loss: 3.2035 - val_accuracy: 0.5782 - val_loss: 3.1142
Epoch 4/50
14/14            3s 114ms/step -
accuracy: 0.6048 - loss: 3.0829 - val_accuracy: 0.5782 - val_loss: 2.9997
Epoch 5/50
14/14            2s 116ms/step -
accuracy: 0.6016 - loss: 2.9689 - val_accuracy: 0.5782 - val_loss: 2.8907
Epoch 6/50
14/14            3s 184ms/step -
accuracy: 0.6038 - loss: 2.8629 - val_accuracy: 0.5782 - val_loss: 2.7872
Epoch 7/50
14/14            3s 192ms/step -
accuracy: 0.5966 - loss: 2.7609 - val_accuracy: 0.5782 - val_loss: 2.6890
Epoch 8/50
```

```
14/14            2s 114ms/step -
accuracy: 0.5976 - loss: 2.6615 - val_accuracy: 0.5782 - val_loss: 2.5956
Epoch 9/50
14/14            2s 112ms/step -
accuracy: 0.6021 - loss: 2.5714 - val_accuracy: 0.5782 - val_loss: 2.5070
Epoch 10/50
14/14            2s 112ms/step -
accuracy: 0.6041 - loss: 2.4825 - val_accuracy: 0.5782 - val_loss: 2.4224
Epoch 11/50
14/14            3s 112ms/step -
accuracy: 0.6014 - loss: 2.3971 - val_accuracy: 0.5782 - val_loss: 2.3422
Epoch 12/50
14/14            3s 114ms/step -
accuracy: 0.6000 - loss: 2.3201 - val_accuracy: 0.5782 - val_loss: 2.2658
Epoch 13/50
14/14            3s 169ms/step -
accuracy: 0.6024 - loss: 2.2443 - val_accuracy: 0.5782 - val_loss: 2.1930
Epoch 14/50
14/14            2s 144ms/step -
accuracy: 0.6064 - loss: 2.1730 - val_accuracy: 0.5782 - val_loss: 2.1239
Epoch 15/50
14/14            2s 111ms/step -
accuracy: 0.5982 - loss: 2.1045 - val_accuracy: 0.5782 - val_loss: 2.0584
Epoch 16/50
14/14            3s 117ms/step -
accuracy: 0.6068 - loss: 2.0380 - val_accuracy: 0.5782 - val_loss: 1.9960
Epoch 17/50
14/14            3s 117ms/step -
accuracy: 0.5997 - loss: 1.9784 - val_accuracy: 0.5782 - val_loss: 1.9367
Epoch 18/50
14/14            2s 113ms/step -
accuracy: 0.5999 - loss: 1.9165 - val_accuracy: 0.5782 - val_loss: 1.8802
Epoch 19/50
14/14            2s 171ms/step -
accuracy: 0.5995 - loss: 1.8595 - val_accuracy: 0.5782 - val_loss: 1.8265
Epoch 20/50
14/14            3s 175ms/step -
accuracy: 0.5998 - loss: 1.8066 - val_accuracy: 0.5782 - val_loss: 1.7754
Epoch 21/50
14/14            2s 115ms/step -
accuracy: 0.5998 - loss: 1.7550 - val_accuracy: 0.5782 - val_loss: 1.7269
Epoch 22/50
14/14            2s 117ms/step -
accuracy: 0.5989 - loss: 1.7091 - val_accuracy: 0.5782 - val_loss: 1.6807
Epoch 23/50
14/14            2s 117ms/step -
accuracy: 0.6022 - loss: 1.6640 - val_accuracy: 0.5782 - val_loss: 1.6366
Epoch 24/50
```

```
14/14            2s 114ms/step -
accuracy: 0.6012 - loss: 1.6242 - val_accuracy: 0.5782 - val_loss: 1.5947
Epoch 25/50
14/14            2s 111ms/step -
accuracy: 0.5989 - loss: 1.5791 - val_accuracy: 0.5782 - val_loss: 1.5548
Epoch 26/50
14/14            2s 111ms/step -
accuracy: 0.6072 - loss: 1.5400 - val_accuracy: 0.5782 - val_loss: 1.5168
Epoch 27/50
14/14            2s 158ms/step -
accuracy: 0.6031 - loss: 1.5003 - val_accuracy: 0.5782 - val_loss: 1.4805
Epoch 28/50
14/14            3s 176ms/step -
accuracy: 0.6025 - loss: 1.4647 - val_accuracy: 0.5782 - val_loss: 1.4458
Epoch 29/50
14/14            2s 114ms/step -
accuracy: 0.6027 - loss: 1.4327 - val_accuracy: 0.5782 - val_loss: 1.4126
Epoch 30/50
14/14            2s 117ms/step -
accuracy: 0.6066 - loss: 1.3964 - val_accuracy: 0.5782 - val_loss: 1.3805
Epoch 31/50
14/14            2s 117ms/step -
accuracy: 0.6128 - loss: 1.3615 - val_accuracy: 0.5782 - val_loss: 1.3500
Epoch 32/50
14/14            3s 114ms/step -
accuracy: 0.6197 - loss: 1.3359 - val_accuracy: 0.5782 - val_loss: 1.3200
Epoch 33/50
14/14            2s 115ms/step -
accuracy: 0.6204 - loss: 1.3113 - val_accuracy: 0.5782 - val_loss: 1.2907
Epoch 34/50
14/14            3s 171ms/step -
accuracy: 0.6544 - loss: 1.2764 - val_accuracy: 0.5877 - val_loss: 1.2620
Epoch 35/50
14/14            3s 182ms/step -
accuracy: 0.7094 - loss: 1.2427 - val_accuracy: 0.6019 - val_loss: 1.2331
Epoch 36/50
14/14            4s 111ms/step -
accuracy: 0.7292 - loss: 1.2125 - val_accuracy: 0.6540 - val_loss: 1.2036
Epoch 37/50
14/14            3s 112ms/step -
accuracy: 0.7756 - loss: 1.1734 - val_accuracy: 0.7062 - val_loss: 1.1718
Epoch 38/50
14/14            3s 116ms/step -
accuracy: 0.8125 - loss: 1.1429 - val_accuracy: 0.8294 - val_loss: 1.1375
Epoch 39/50
14/14            3s 181ms/step -
accuracy: 0.8559 - loss: 1.0959 - val_accuracy: 0.8483 - val_loss: 1.0997
Epoch 40/50
```

```
14/14              3s 184ms/step -
accuracy: 0.8913 - loss: 1.0554 - val_accuracy: 0.8720 - val_loss: 1.0584
Epoch 41/50
14/14              4s 113ms/step -
accuracy: 0.9179 - loss: 0.9913 - val_accuracy: 0.8815 - val_loss: 1.0142
Epoch 42/50
14/14              2s 113ms/step -
accuracy: 0.9273 - loss: 0.9354 - val_accuracy: 0.8910 - val_loss: 0.9691
Epoch 43/50
14/14              3s 118ms/step -
accuracy: 0.9284 - loss: 0.8721 - val_accuracy: 0.8863 - val_loss: 0.9289
Epoch 44/50
14/14              2s 110ms/step -
accuracy: 0.9412 - loss: 0.8235 - val_accuracy: 0.8910 - val_loss: 0.8955
Epoch 45/50
14/14              2s 169ms/step -
accuracy: 0.9519 - loss: 0.7766 - val_accuracy: 0.8957 - val_loss: 0.8697
Epoch 46/50
14/14              3s 187ms/step -
accuracy: 0.9556 - loss: 0.7446 - val_accuracy: 0.8910 - val_loss: 0.8468
Epoch 47/50
14/14              2s 114ms/step -
accuracy: 0.9519 - loss: 0.7133 - val_accuracy: 0.8910 - val_loss: 0.8295
Epoch 48/50
14/14              2s 112ms/step -
accuracy: 0.9625 - loss: 0.7010 - val_accuracy: 0.8910 - val_loss: 0.8131
Epoch 49/50
14/14              2s 113ms/step -
accuracy: 0.9626 - loss: 0.6633 - val_accuracy: 0.8910 - val_loss: 0.7984
Epoch 50/50
14/14              2s 113ms/step -
accuracy: 0.9698 - loss: 0.6470 - val_accuracy: 0.8863 - val_loss: 0.7850
7/7              0s 29ms/step
Fold 4 - Precision: 0.8769, Recall: 0.9344, F1 Score: 0.9048
Training for fold 5 on PolitiFact…
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

14/14              6s 201ms/step -
accuracy: 0.5680 - loss: 3.4520 - val_accuracy: 0.5857 - val_loss: 3.3528
Epoch 2/50
14/14              2s 115ms/step -
accuracy: 0.5995 - loss: 3.3196 - val_accuracy: 0.5857 - val_loss: 3.2278
Epoch 3/50
14/14              3s 117ms/step -
accuracy: 0.6112 - loss: 3.1962 - val_accuracy: 0.5857 - val_loss: 3.1090
```

```
Epoch 4/50
14/14            2s 112ms/step -
accuracy: 0.6152 - loss: 3.0774 - val_accuracy: 0.5857 - val_loss: 2.9966
Epoch 5/50
14/14            3s 118ms/step -
accuracy: 0.6066 - loss: 2.9667 - val_accuracy: 0.5857 - val_loss: 2.8898
Epoch 6/50
14/14            2s 113ms/step -
accuracy: 0.6172 - loss: 2.8593 - val_accuracy: 0.5857 - val_loss: 2.7881
Epoch 7/50
14/14            2s 175ms/step -
accuracy: 0.6086 - loss: 2.7582 - val_accuracy: 0.5857 - val_loss: 2.6914
Epoch 8/50
14/14            3s 188ms/step -
accuracy: 0.6123 - loss: 2.6621 - val_accuracy: 0.5857 - val_loss: 2.5995
Epoch 9/50
14/14            2s 112ms/step -
accuracy: 0.6211 - loss: 2.5705 - val_accuracy: 0.5857 - val_loss: 2.5121
Epoch 10/50
14/14            2s 112ms/step -
accuracy: 0.6219 - loss: 2.4845 - val_accuracy: 0.5857 - val_loss: 2.4290
Epoch 11/50
14/14            2s 117ms/step -
accuracy: 0.6151 - loss: 2.4029 - val_accuracy: 0.5857 - val_loss: 2.3498
Epoch 12/50
14/14            2s 113ms/step -
accuracy: 0.6224 - loss: 2.3226 - val_accuracy: 0.5857 - val_loss: 2.2744
Epoch 13/50
14/14            2s 117ms/step -
accuracy: 0.6147 - loss: 2.2477 - val_accuracy: 0.5857 - val_loss: 2.2028
Epoch 14/50
14/14            2s 166ms/step -
accuracy: 0.6256 - loss: 2.1761 - val_accuracy: 0.5857 - val_loss: 2.1344
Epoch 15/50
14/14            3s 184ms/step -
accuracy: 0.6243 - loss: 2.1076 - val_accuracy: 0.5857 - val_loss: 2.0694
Epoch 16/50
14/14            4s 112ms/step -
accuracy: 0.6103 - loss: 2.0447 - val_accuracy: 0.5857 - val_loss: 2.0076
Epoch 17/50
14/14            2s 116ms/step -
accuracy: 0.6370 - loss: 1.9819 - val_accuracy: 0.5857 - val_loss: 1.9485
Epoch 18/50
14/14            2s 114ms/step -
accuracy: 0.6400 - loss: 1.9225 - val_accuracy: 0.5857 - val_loss: 1.8925
Epoch 19/50
14/14            3s 113ms/step -
accuracy: 0.6413 - loss: 1.8665 - val_accuracy: 0.5857 - val_loss: 1.8390
```

```
Epoch 20/50
14/14            3s 173ms/step -
accuracy: 0.6402 - loss: 1.8106 - val_accuracy: 0.5857 - val_loss: 1.7879
Epoch 21/50
14/14            2s 163ms/step -
accuracy: 0.6702 - loss: 1.7611 - val_accuracy: 0.5857 - val_loss: 1.7391
Epoch 22/50
14/14            2s 112ms/step -
accuracy: 0.6766 - loss: 1.7099 - val_accuracy: 0.5905 - val_loss: 1.6925
Epoch 23/50
14/14            2s 112ms/step -
accuracy: 0.6696 - loss: 1.6643 - val_accuracy: 0.5905 - val_loss: 1.6478
Epoch 24/50
14/14            3s 117ms/step -
accuracy: 0.7093 - loss: 1.6142 - val_accuracy: 0.5905 - val_loss: 1.6050
Epoch 25/50
14/14            3s 150ms/step -
accuracy: 0.7123 - loss: 1.5725 - val_accuracy: 0.6286 - val_loss: 1.5640
Epoch 26/50
14/14            3s 193ms/step -
accuracy: 0.7299 - loss: 1.5256 - val_accuracy: 0.6667 - val_loss: 1.5241
Epoch 27/50
14/14            4s 110ms/step -
accuracy: 0.7604 - loss: 1.4823 - val_accuracy: 0.7048 - val_loss: 1.4853
Epoch 28/50
14/14            3s 112ms/step -
accuracy: 0.7941 - loss: 1.4365 - val_accuracy: 0.7238 - val_loss: 1.4474
Epoch 29/50
14/14            2s 114ms/step -
accuracy: 0.8418 - loss: 1.3912 - val_accuracy: 0.7476 - val_loss: 1.4097
Epoch 30/50
14/14            2s 113ms/step -
accuracy: 0.8379 - loss: 1.3507 - val_accuracy: 0.7429 - val_loss: 1.3723
Epoch 31/50
14/14            2s 112ms/step -
accuracy: 0.8492 - loss: 1.3024 - val_accuracy: 0.7571 - val_loss: 1.3344
Epoch 32/50
14/14            2s 131ms/step -
accuracy: 0.8806 - loss: 1.2474 - val_accuracy: 0.7667 - val_loss: 1.2963
Epoch 33/50
14/14            3s 185ms/step -
accuracy: 0.8943 - loss: 1.1923 - val_accuracy: 0.7714 - val_loss: 1.2584
Epoch 34/50
14/14            2s 119ms/step -
accuracy: 0.9179 - loss: 1.1270 - val_accuracy: 0.7810 - val_loss: 1.2221
Epoch 35/50
14/14            2s 115ms/step -
accuracy: 0.9255 - loss: 1.0804 - val_accuracy: 0.7857 - val_loss: 1.1890
```

```
Epoch 36/50
14/14            2s 117ms/step -
accuracy: 0.9325 - loss: 1.0235 - val_accuracy: 0.7905 - val_loss: 1.1583
Epoch 37/50
14/14            2s 119ms/step -
accuracy: 0.9487 - loss: 0.9690 - val_accuracy: 0.8000 - val_loss: 1.1303
Epoch 38/50
14/14            2s 113ms/step -
accuracy: 0.9401 - loss: 0.9277 - val_accuracy: 0.8048 - val_loss: 1.1058
Epoch 39/50
14/14            2s 114ms/step -
accuracy: 0.9494 - loss: 0.8895 - val_accuracy: 0.8095 - val_loss: 1.0840
Epoch 40/50
14/14            3s 175ms/step -
accuracy: 0.9522 - loss: 0.8527 - val_accuracy: 0.8048 - val_loss: 1.0640
Epoch 41/50
14/14            2s 162ms/step -
accuracy: 0.9557 - loss: 0.8343 - val_accuracy: 0.8190 - val_loss: 1.0457
Epoch 42/50
14/14            2s 114ms/step -
accuracy: 0.9627 - loss: 0.7917 - val_accuracy: 0.8429 - val_loss: 1.0323
Epoch 43/50
14/14            2s 113ms/step -
accuracy: 0.9655 - loss: 0.7819 - val_accuracy: 0.8429 - val_loss: 1.0170
Epoch 44/50
14/14            2s 115ms/step -
accuracy: 0.9699 - loss: 0.7566 - val_accuracy: 0.8571 - val_loss: 0.9992
Epoch 45/50
14/14            3s 114ms/step -
accuracy: 0.9701 - loss: 0.7284 - val_accuracy: 0.8524 - val_loss: 0.9837
Epoch 46/50
14/14            3s 124ms/step -
accuracy: 0.9622 - loss: 0.7089 - val_accuracy: 0.8524 - val_loss: 0.9705
Epoch 47/50
14/14            2s 172ms/step -
accuracy: 0.9696 - loss: 0.6897 - val_accuracy: 0.8524 - val_loss: 0.9560
Epoch 48/50
14/14            3s 177ms/step -
accuracy: 0.9685 - loss: 0.6736 - val_accuracy: 0.8571 - val_loss: 0.9459
Epoch 49/50
14/14            2s 113ms/step -
accuracy: 0.9681 - loss: 0.6585 - val_accuracy: 0.8571 - val_loss: 0.9383
Epoch 50/50
14/14            2s 118ms/step -
accuracy: 0.9745 - loss: 0.6397 - val_accuracy: 0.8571 - val_loss: 0.9239
7/7            0s 29ms/step
Fold 5 - Precision: 0.8908, Recall: 0.8618, F1 Score: 0.8760
Average Precision: 0.8684
```

```
Average Recall: 0.8965
Average F1 Score: 0.8817
```

The metrics across different folds show stable performance for both datasets:

GossipCop

```
Fold 1 - Precision: 0.8644, Recall: 0.9141, F1 Score: 0.8885
Fold 2 - Precision: 0.8703, Recall: 0.9051, F1 Score: 0.8873
Fold 3 - Precision: 0.7545, Recall: 1.0000, F1 Score: 0.8601
Fold 4 - Precision: 0.7642, Recall: 1.0000, F1 Score: 0.8664
Fold 5 - Precision: 0.8675, Recall: 0.8867, F1 Score: 0.8770

Average Precision: 0.8242
Average Recall: 0.9412
Average F1 Score: 0.8759
```

PolitiFact

```
Fold 1 - Precision: 0.8472, Recall: 0.9385, F1 Score: 0.8905
Fold 2 - Precision: 0.8976, Recall: 0.8906, F1 Score: 0.8941
Fold 3 - Precision: 0.8293, Recall: 0.8571, F1 Score: 0.8430
Fold 4 - Precision: 0.8769, Recall: 0.9344, F1 Score: 0.9048
Fold 5 - Precision: 0.8908, Recall: 0.8618, F1 Score: 0.8760

Average Precision: 0.8648
Average Recall: 0.8965
Average F1 Score: 0.8817
```

1.10 Hyperparameter Tuning

I have selected Keras tuner for my CNN hyperparameter tuning. Keras Tuner demonstrates superior accuracy in CNN applications (see Table II in Halim et al.[18]).

1.10.1 Hyperparameter Tuning Functions

[17]:

```
# This class defines a hypermodel for hyperparameter tuning with Keras Tuner
# It accepts hyperparameters to tune and inherits from kt.HyperModel and allows␣
↪integration with the Keras Tuner library
class CNNHyperModel(kt.HyperModel):
    def __init__(self, input_dim, max_len):
        # Initialize the hypermodel with the maximum vocabulary size and the␣
↪maximum length of input sequences
        self.input_dim = input_dim
        self.max_len = max_len

    def build(self, hp):
        # Build the model architecture with hyperparameters that will be tuned.
        # This method is called by the tuner to create new models with␣
↪different hyperparameter values
```

```
        model = Sequential([
            # Embedding layer to transform indices into dense vectors of fixed␣
 ↪size
            Embedding(input_dim=self.input_dim, output_dim=128,␣
 ↪input_length=self.max_len),
            # Conv1D layer with hyperparameters for number of filters and␣
 ↪kernel size that will be tuned
            Conv1D(
                filters=hp.Choice('filters', [32, 64, 128]),
                kernel_size=hp.Choice('kernel_size', [3, 5]),
                activation='relu',
                kernel_regularizer=l2(0.01)  # L2 regularization to prevent␣
 ↪overfitting.
            ),
            MaxPooling1D(pool_size=2),  # MaxPooling to reduce the spatial␣
 ↪dimensions
            GlobalMaxPooling1D(),  # Global max pooling to reduce the output of␣
 ↪the convolutional layer to a single vector
            Dense(
                units=hp.Int('dense_units', min_value=32, max_value=128,␣
 ↪step=32),
                activation='relu'
            ),
            Dropout(rate=hp.Float('dropout_rate', min_value=0.0, max_value=0.5,␣
 ↪step=0.1)),  # Dropout layer to prevent overfitting
            Dense(1, activation='sigmoid')  # Output layer with sigmoid␣
 ↪activation for binary classification
        ])
        model.compile(
            optimizer=Adam(hp.Float('learning_rate', min_value=1e-4,␣
 ↪max_value=1e-2, sampling='LOG')),
            loss='binary_crossentropy',
            metrics=[tf.keras.metrics.AUC(curve='PR')]  # Use Precision-Recall␣
 ↪AUC as a metric for imbalanced classification
        )
        return model

# This function configures and executes the hyperparameter tuning using Keras␣
 ↪Tuner
# It sets up the tuner, defines the hypermodel, and manages the training process
# Function includes callbacks for early stopping and learning rate reduction
def tune_model(X_train, y_train, X_test, y_test, input_dim, max_len, n_trials,␣
 ↪directory, project_name):
    # Setup the hypermodel instance with provided input dimension and maximum␣
 ↪length
    hypermodel = CNNHyperModel(input_dim=input_dim, max_len=max_len)
```

```
    # Configure the tuner with RandomSearch algorithm, focus on maximizing the␣
↪AUC
    tuner = kt.RandomSearch(
        hypermodel,
        objective=kt.Objective("val_auc", direction="max"),
        max_trials=n_trials,
        executions_per_trial=1,
        directory=directory,
        project_name=project_name
    )

    # Validation data used by tuner
    val_data = (X_test, y_test)
    # Use existing F1 score callback for additional performance metric tracking
    f1_callback = F1ScoreCallback(validation_data=val_data)

    # Start hyperparameter search process, use early stopping and learning rate␣
↪reduction to optimize training
    tuner.search(
        x=X_train,
        y=y_train,
        epochs=50,
        validation_data=val_data,
        callbacks=[EarlyStopping(monitor='val_loss', patience=5,␣
↪restore_best_weights=True),
                   ReduceLROnPlateau(monitor='val_loss', factor=0.1,␣
↪patience=2, min_lr=0.00001),
                   f1_callback]
    )

    # Get the best model after the search
    best_model = tuner.get_best_models(num_models=1)[0]
    return tuner, best_model  # Return the tuner and the best model
```

1.10.2 Hyperparameter Tuning Results

[18]:

```
# Run hyperparameter tuning for each dataset
tuner_gossipcop, best_model_gossipcop = tune_model(
    X_train_gossipcop, y_train_gossipcop, X_test_gossipcop, y_test_gossipcop,
    5000, 100, 10, 'hyper_tuning_gossipcop', 'GossipCopTuning'
)

tuner_politifact, best_model_politifact = tune_model(
    X_train_politifact, y_train_politifact, X_test_politifact,␣
↪y_test_politifact,
```

```
    5000, 100, 10, 'hyper_tuning_politifact', 'PolitiFactTuning'
)
```

```
Trial 10 Complete [00h 00m 16s]
val_auc: 0.9387391209602356

Best val_auc So Far: 0.9391252994537354
Total elapsed time: 00h 06m 14s
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/keras/src/saving/saving_lib.py:576:
UserWarning: Skipping variable loading for optimizer 'adam', because it has 2
variables whereas the saved optimizer has 16 variables.
  saveable.load_own_variables(weights_store.get(inner_path))
```

[19]:

```python
# This function displays the best hyperparameters found by the tuner for a␣
 ↪specific dataset
def display_best_hyperparameters_and_summary(tuner, dataset_name):
    # Get the best hyperparameters
    best_hps = tuner.get_best_hyperparameters()[0]
    print(f"Best hyperparameters for {dataset_name}:")
    print(f"  Best filter size: {best_hps.get('filters')}")
    print(f"  Best kernel size: {best_hps.get('kernel_size')}")
    print(f"  Best dense units: {best_hps.get('dense_units')}")
    print(f"  Best learning rate: {best_hps.get('learning_rate')}\n")

    # Get the best model and print the model summary
    best_model = tuner.get_best_models(num_models=1)[0]
    print(f"Model summary for {dataset_name}:")
    best_model.summary()

# GossipCop tuning results
display_best_hyperparameters_and_summary(tuner_gossipcop, "GossipCop")

# Politifact tuning results
display_best_hyperparameters_and_summary(tuner_politifact, "PolitiFact")
```

```
Best hyperparameters for GossipCop:
  Best filter size: 32
  Best kernel size: 5
  Best dense units: 32
  Best learning rate: 0.0001655876301640272

Model summary for GossipCop:
```

Model: "sequential"

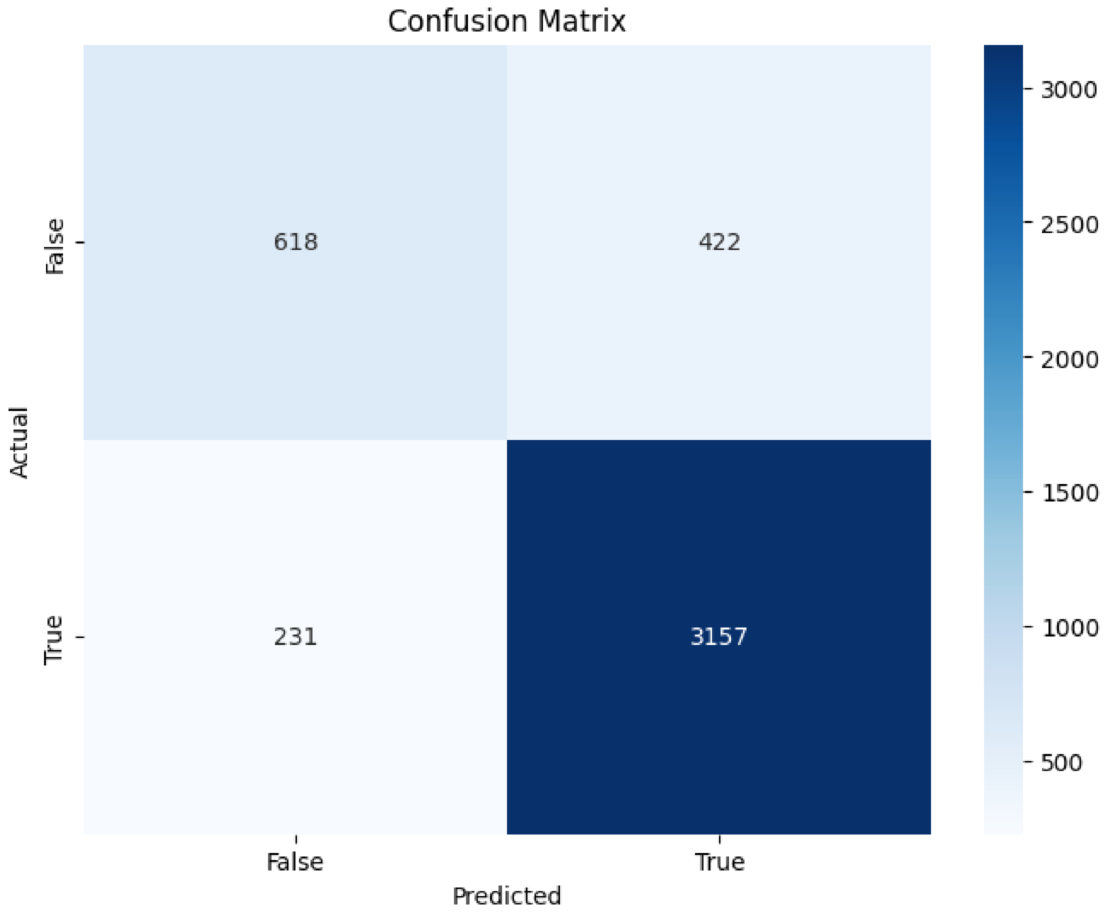
```
 Layer (type)                         Output Shape                         ␣
 ↪Param #

 embedding (Embedding)                (None, 100, 128)                     ␣
 ↪640,000

 conv1d (Conv1D)                      (None, 96, 32)                        ␣
 ↪20,512

 max_pooling1d (MaxPooling1D)         (None, 48, 32)                           ␣
 ↪  0

 global_max_pooling1d                 (None, 32)                               ␣
 ↪  0
 (GlobalMaxPooling1D)                                                          ␣
 ↪

 dense (Dense)                        (None, 32)                             ␣
 ↪1,056

 dropout (Dropout)                    (None, 32)                               ␣
 ↪  0

 dense_1 (Dense)                      (None, 1)                                ␣
 ↪ 33


 Total params: 661,601 (2.52 MB)

 Trainable params: 661,601 (2.52 MB)

 Non-trainable params: 0 (0.00 B)

Best hyperparameters for PolitiFact:
  Best filter size: 32
  Best kernel size: 5
  Best dense units: 64
  Best learning rate: 0.0011563133814945622

Model summary for PolitiFact:

Model: "sequential"
```

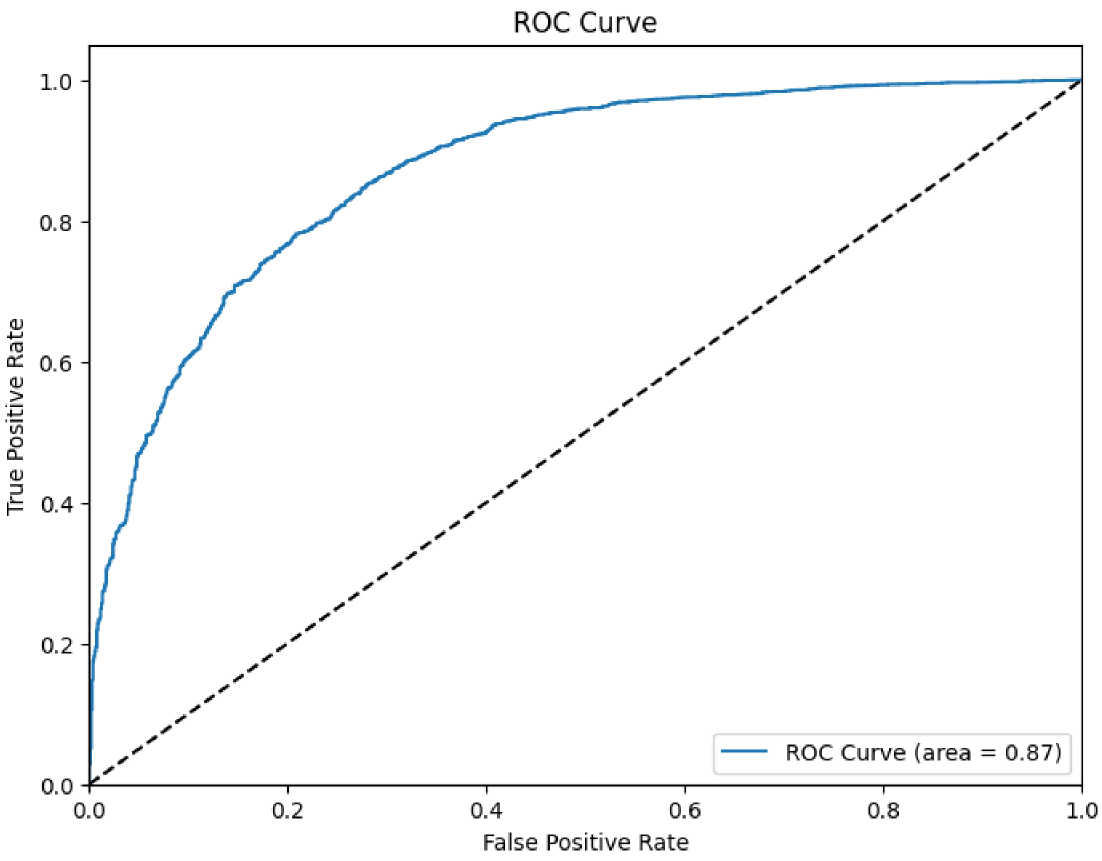
```
Layout (type)                          Output Shape                          ␣
↪Param #

embedding (Embedding)                  (None, 100, 128)                      ␣
↪640,000

conv1d (Conv1D)                        (None, 96, 32)                        ␣
↪20,512

max_pooling1d (MaxPooling1D)           (None, 48, 32)                          ␣
↪  0

global_max_pooling1d                   (None, 32)                              ␣
↪  0
(GlobalMaxPooling1D)                                                           ␣
↪

dense (Dense)                          (None, 64)                             ␣
↪2,112

dropout (Dropout)                      (None, 64)                              ␣
↪  0

dense_1 (Dense)                        (None, 1)                               ␣
↪ 65


Total params: 662,689 (2.53 MB)

Trainable params: 662,689 (2.53 MB)

Non-trainable params: 0 (0.00 B)
```

[20]:

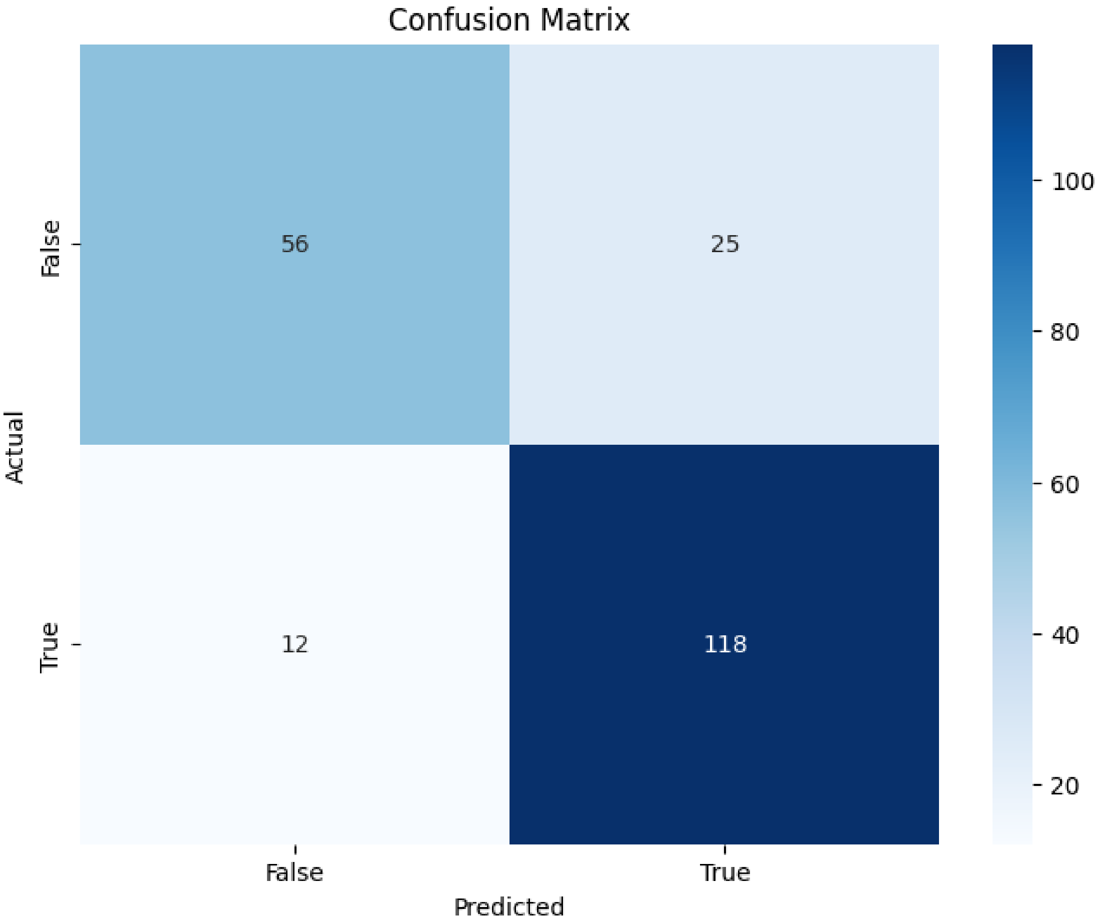
```
# Evaluate the tuned model for GossipCop dataset
_, y_pred_gossipcop_tuned, _ = train_and_evaluate(best_model_gossipcop, None,␣
 ↪X_test_gossipcop, None, y_test_gossipcop, None, None, None,␣
 ↪train_model=False)

# Evaluate the tuned model for PolitiFact dataset
_, y_pred_politifact_tuned, _ = train_and_evaluate(best_model_politifact, None,␣
 ↪X_test_politifact, None, y_test_politifact, None, None, None,␣
 ↪train_model=False)
```

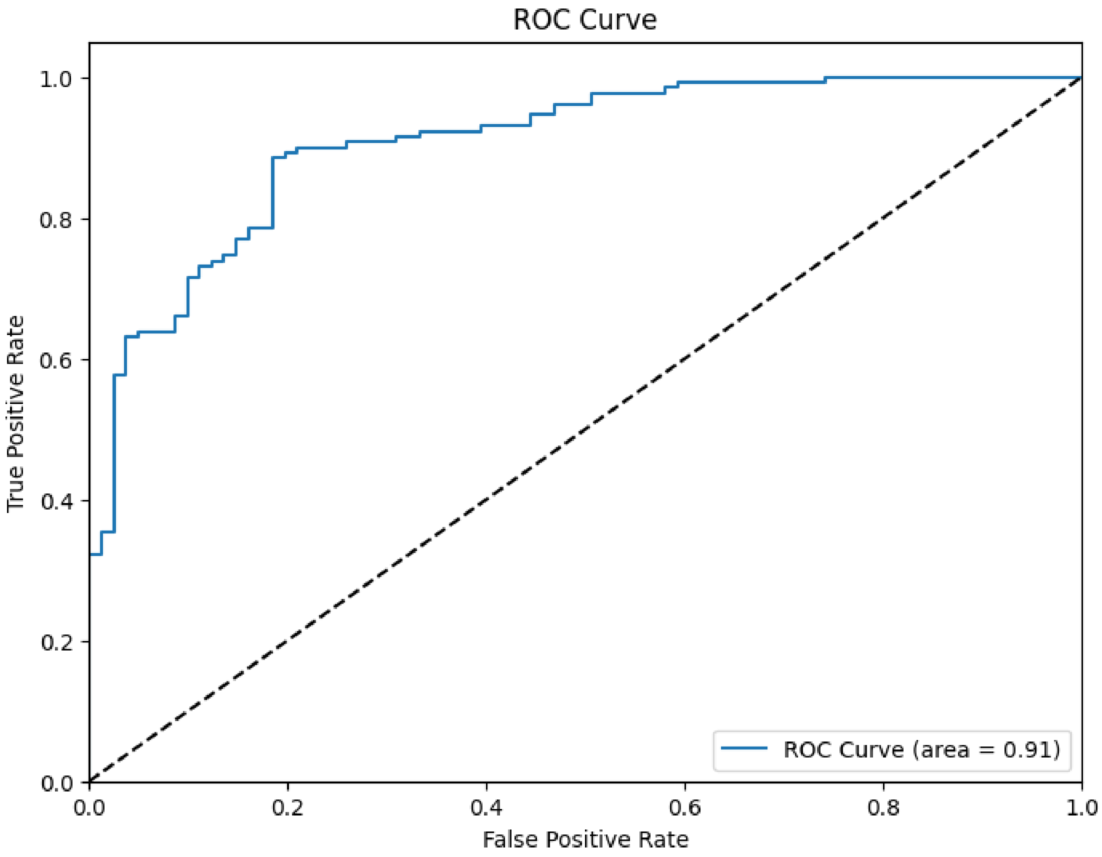
```
139/139             1s 9ms/step
Accuracy: 0.8525
Precision: 0.8821
Recall: 0.9318
F1-score: 0.9063
ROC AUC: 0.8722
Confusion Matrix:
[[ 618  422]
 [ 231 3157]]
```





```
7/7               0s 20ms/step
Accuracy: 0.8246
Precision: 0.8252
Recall: 0.9077
F1-score: 0.8645
ROC AUC: 0.9066
Confusion Matrix:
[[ 56  25]
 [ 12 118]]
```


Confusion Matrix
Actual
False
True
56
25
12
118
Predicted
False
True
100
80
60
40
20



GossipCop Dataset

Initial Model Results:

```
Accuracy: 0.8318
Precision: 0.8948
Recall: 0.8840
F1-score: 0.8894
ROC AUC: 0.8648
```

Tuned Model Results:

```
Accuracy: 0.8525
Precision: 0.8821
Recall: 0.9318
F1-score: 0.9063
ROC AUC: 0.8722
```

Accuracy and ROC AUC show improvements in the tuned model, with better performance. Recall is higher, which means that it is better at identifying actual positive (fake news) instances. Precision has decreased, so the tuned model has a higher rate of false positives. F1-score is higher.

PolitiFact Dataset

Initial Model Results:

```
Accuracy: 0.8057
Precision: 0.8504
Recall: 0.8308
F1-score: 0.8405
ROC AUC: 0.8760
```

Tuned Model Results:

```
Accuracy: 0.8246
Precision: 0.8252
Recall: 0.9077
F1-score: 0.8645
ROC AUC: 0.9066
```

Accuracy has increased in the tuned model, this means it is more effective as the initial model. Precision is lower, but recall has increased, this means the tuned model misses more actual positive cases. F1-score is higher and corresponds to the increase in recall. ROC AUC has improved despite the lower accuracy.

For the GossipCop dataset, the tuned model has improved in most metrics, so this is my preferred model.

For the PolitiFact dataset, the results are mixed. Because of this, I choose the original model and not the tuned model.

1.11 Save the Best Models to Google Drive

[21]:
```python
# Save the best tuned model for GossipCop
best_model_gossipcop = tuner_gossipcop.get_best_models(num_models=1)[0]
model_filename_gossipcop = '/content/drive/My Drive/best_cnn_model_gossipcop.
  ↪keras'
best_model_gossipcop.save(model_filename_gossipcop)
print("Best CNN model for GossipCop saved to Google Drive")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/keras/src/saving/saving_lib.py:576:
UserWarning: Skipping variable loading for optimizer 'adam', because it has 2
variables whereas the saved optimizer has 16 variables.
  saveable.load_own_variables(weights_store.get(inner_path))
```

```
Best CNN model for GossipCop saved to Google Drive
```

[22]:
```python
# Save the GossipCop tokenizer and label encoder
joblib.dump(tokenizer_gossipcop, '/content/drive/My Drive/tokenizer_gossipcop.
  ↪pkl')
joblib.dump(le_gossipcop, '/content/drive/My Drive/label_encoder_gossipcop.pkl')
print("GossipCop tokenizer and label encoder saved to Google Drive")
```

```
GossipCop tokenizer and label encoder saved to Google Drive
```

```
[23]: # Save the best untuned model for PolitiFact
      model_filename_politifact = '/content/drive/My Drive/best_cnn_model_politifact.
       ↪keras'
      cnn_model_politifact.save(model_filename_politifact)
      print("Best CNN model for PolitiFact saved to Google Drive")
```

```
Best CNN model for PolitiFact saved to Google Drive
```

```
[24]: # Save the PolitiFact tokenizer and label encoder
      joblib.dump(tokenizer_politifact, '/content/drive/My Drive/tokenizer_politifact.
       ↪pkl')
      joblib.dump(le_politifact, '/content/drive/My Drive/label_encoder_politifact.
       ↪pkl')
      print("PolitiFact tokenizer and label encoder saved to Google Drive")
```

```
PolitiFact tokenizer and label encoder saved to Google Drive
```

1.12 Conclusion

My models significantly outperform the baseline results across all metrics for both datasets (GossipCop and PolitiFact). Calculated class weights in both datasets handle class imbalance. Hyperparameter tuning improved GosspiCop model performance, which was not the case for PolitiFact model. The best performing model was GossipCop with tuned hyperparameters.

1.13 References

[7] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. https://doi.org/10.48550/arXiv.1809.01286

[11] BRITZ, D. 2021. Convolutional Neural Network for Text Classification in TensorFlow. https://github.com/dennybritz/cnn-text-classification-tf

[15] BROWNE, M.W. 2000. Cross-Validation Methods. Journal of Mathematical Psychology, 44, 108–132. https://doi.org/10.1006/jmps.1999.1279

[16] POWERS, D.M.W. 2011. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. International Journal of Machine Learning Technology, 2(1), pp. 37-63. https://doi.org/10.48550/arXiv.2010.16061

[18] HALIM, A., CHOW, C., BUDIHARTO, M., ACHMAD, S., and SUTOYO, R. 2023. The Impact of Hyperparameter Tuning in Convolutional Neural Network on Image Classification Model: A Case Study of Plant Disease Detection. In Proceedings of the IEEE Conference on Innovative Research and Development (ICORIS), pp. 1-6. https://doi.org/10.1109/ICORIS60118.2023.10352209

```
[ ]: # Install LaTeX packages necessary for converting notebooks to PDF
     !apt-get update
```

```
!apt-get install -y texlive-xetex texlive-fonts-recommended␣
↪texlive-plain-generic texlive-latex-extra pandoc

# Convert the notebook to PDF
!jupyter nbconvert --to pdf "/content/drive/My Drive/Colab Notebooks/
↪FakeNewsNetCNN.ipynb"
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,626 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease [1,581 B]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Get:6 https://r2u.stat.illinois.edu/ubuntu jammy Release [5,713 B]
Get:7 https://r2u.stat.illinois.edu/ubuntu jammy Release.gpg [793 B]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
Packages [976 kB]
Get:10 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,287 kB]
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,499
kB]
Get:13 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,569 kB]
Get:14 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
[18.1 kB]
Hit:15 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Hit:16 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:17 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy/main amd64
Packages [27.8 kB]
Fetched 14.6 MB in 3s (4,836 kB/s)
Reading package lists… Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
```

```
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc-data
poppler-data
  preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick ruby-
xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base texlive-
binaries
  texlive-latex-base texlive-latex-recommended texlive-pictures tipa xfonts-
encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java texlive-
luatex
  pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc nodejs php python
libjs-mathjax
  libjs-katex citation-style-language-styles poppler-utils ghostscript fonts-
japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-
arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv | postscript-
viewer perl-tk xpdf
  | pdf-viewer xzdec texlive-fonts-recommended-doc texlive-latex-base-doc
python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-
extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex texlive-
pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc
pandoc-data
  poppler-data preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-
webrick ruby-xmlrpc
  ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
texlive-binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-extra texlive-
latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa xfonts-encodings
xfonts-utils
0 upgraded, 58 newly installed, 0 to remove and 49 not upgraded.
```

```
Need to get 202 MB of archives.
After this operation, 728 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.9 [752 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.9 [5,033 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-
gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-gfm-
extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
```

```
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.7 [50.1 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
Get:28 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ruby-webrick all
1.7.0-3 [51.8 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.7 [5,113 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc-data all
2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc amd64
2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
```

```
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 12s (17.3 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 123597 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.9_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.9) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
```

```
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.9_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.9) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack …/17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack …/18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/19-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/20-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/21-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
```

```
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/23-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/24-ruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/25-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/26-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/27-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/28-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/29-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/31-libruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/32-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/33-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/36-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
```

```
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/38-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/39-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package pandoc-data.
Preparing to unpack …/40-pandoc-data_2.9.2.1-3ubuntu2_all.deb …
Unpacking pandoc-data (2.9.2.1-3ubuntu2) …
Selecting previously unselected package pandoc.
Preparing to unpack …/41-pandoc_2.9.2.1-3ubuntu2_amd64.deb …
Unpacking pandoc (2.9.2.1-3ubuntu2) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/42-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/43-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/44-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/45-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/47-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/48-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/49-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/50-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/51-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
```

```
Preparing to unpack …/52-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/53-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/54-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/55-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/56-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/57-texlive-xetex_2021.20220204-1_all.deb …
Unpacking texlive-xetex (2021.20220204-1) …
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
Setting up tex-common (6.17) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up ruby-webrick (1.7.0-3) …
Setting up libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up pandoc-data (2.9.2.1-3ubuntu2) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
```

```
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.9) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.9) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up pandoc (2.9.2.1-3ubuntu2) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
Setting up tipa (2:1.3-21) …
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Setting up ruby3.0 (3.0.2-7ubuntu2.7) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
```

```
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.4) …
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a
symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

Processing triggers for tex-common (6.17) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time…
```

FakeNewsNetInference

September 8, 2024

1 FakeNewsNetInference

This notebook helps users distinguish between fake and real news titles using CNN models created in the FakeNewsNetCNN notebook. The models were trained on the FakeNewsNet dataset, the GossipCop and PoliticFact subsets. For more information, please see the project folder, start with the Report.pdf.

1.1 Instructions for Use

- **Type the news title** into the text area below and select the type of news (Gossip or Political) from the dropdown menu.
- Press the "Check Fake News" button.
- The model will predict and display whether the news is likely **Fake** or **Not Fake** according to the selected dataset.
- To show or hide the code, click the arrow (>) on the left side of the "Inference code" cell.
- Please note: The models work best with English text.

1.2 Handling Disconnections

- If the Colab runtime disconnects (noticeable at the top-right corner of the page), please reconnect by clicking the 'Connect' button.
- After reconnecting, re-run the cell containing the Inference code.

[8]:

```
# @title # Inference code

# Install libraries with specified versions for compatibility of model loading␣
 ↪and import necessary libraries
!pip install tensorflow==2.17.0 gdown==5.1.0 joblib==1.4.2 ipywidgets==7.7.1

import tensorflow as tf
import ipywidgets as widgets
from IPython.display import display, HTML
from tensorflow.keras.preprocessing.sequence import pad_sequences
import joblib
import gdown
import zipfile
import os
import time
import re
```

```
# Public links to files
gossipcop_model_url = 'https://drive.google.com/uc?
 ↪id=1tgL8ITlc61l0BnjDLbnyOJ8zghtNDjVL'
gossipcop_tokenizer_file_url = 'https://drive.google.com/uc?
 ↪id=12e6MIz9qKCbLu6cz7zyRytOVAwKPOMYI'
gossipcop_label_encoder_file_url = 'https://drive.google.com/uc?
 ↪id=1PT-7nFgXT0qISSRqEe9jTMJdAro5aQ3j'
politifact_model_url = 'https://drive.google.com/uc?
 ↪id=14wU7pqr6tlpg3bE89lx3Y-Nf0O_Q4Mwy'
politifact_tokenizer_file_url = 'https://drive.google.com/uc?
 ↪id=19nmqTyZoXZcbc24Gw7lYEKyLNVoKGine'
politifact_label_encoder_file_url = 'https://drive.google.com/uc?
 ↪id=1wHBlUc4xLZYdbkW8yVmmVQtxpKyHAftD'

# Download files from Google Drive
gdown.download(gossipcop_model_url, "best_cnn_model_gossipcop.keras",␣
 ↪quiet=False)
gdown.download(gossipcop_tokenizer_file_url, "tokenizer_gossipcop.pkl",␣
 ↪quiet=False)
gdown.download(gossipcop_label_encoder_file_url, "label_encoder_gossipcop.pkl",␣
 ↪quiet=False)
gdown.download(politifact_model_url, "best_cnn_model_politifact.keras",␣
 ↪quiet=False)
gdown.download(politifact_tokenizer_file_url, "tokenizer_politifact.pkl",␣
 ↪quiet=False)
gdown.download(politifact_label_encoder_file_url, "label_encoder_politifact.
 ↪pkl", quiet=False)

# Load the CNN models
model_gossipcop = tf.keras.models.load_model('best_cnn_model_gossipcop.keras')
model_politifact = tf.keras.models.load_model('best_cnn_model_politifact.keras')

# Load tokenizers and label encoders
tokenizer_gossipcop = joblib.load('tokenizer_gossipcop.pkl')
le_gossipcop = joblib.load('label_encoder_gossipcop.pkl')
tokenizer_politifact = joblib.load('tokenizer_politifact.pkl')
le_politifact = joblib.load('label_encoder_politifact.pkl')

print("Setup complete!")

# Function to preprocess text
def preprocess_text(text, tokenizer):
    seq = tokenizer.texts_to_sequences([text])
    padded = pad_sequences(seq, maxlen=100)
    return padded
```

```python
# Define widgets for user input
text_input = widgets.Textarea(
    value='',
    placeholder='Type the news title here...',
    description='News Title:',
    disabled=False,
    layout=widgets.Layout(width='100%', height='200px')
)

news_type = widgets.Dropdown(
    options=['Gossip', 'Political'],
    value='Gossip',
    description='News Type:',
    disabled=False,
)

button = widgets.Button(description="Check Fake News", tooltip="Click to check if the news is fake or not")
clear_button = widgets.Button(description="Clear", tooltip="Clear the input text")
output = widgets.Output()
progress = widgets.IntProgress(value=0, min=0, max=100, step=10, description='Processing:', bar_style='info', style={'bar_color': '#00BFFF'})

# Function to validate input text
def validate_input(text):
    if not text.strip():
        return False, "Input is empty. Please enter a news title."
    if len(re.findall(r'\w+', text)) < 3:  # Check for minimum number of words
        return False, "Input is too short or not meaningful. Please enter a valid news title."
    return True, ""

# Precompile the prediction functions
gossipcop_predict_fn = model_gossipcop.make_predict_function()
politifact_predict_fn = model_politifact.make_predict_function()

# Button click event handler
def on_button_clicked(b):
    with output:
        output.clear_output()
        progress.value = 0
        input_text = text_input.value
        is_valid, message = validate_input(input_text)
        if not is_valid:
```

```
            display(HTML(f"<p style='color: red; font-size: 16px;'>{message}</
↪p>"))
            return

        progress.value = 20
        display(progress)
        for i in range(8):
            time.sleep(0.1)
            progress.value += 10

        news_type_selected = news_type.value

        if news_type_selected == 'Gossip':
            # Preprocess the text
            preprocessed_text = preprocess_text(input_text, tokenizer_gossipcop)
            progress.value = 90
            # Making predictions
            prediction = model_gossipcop(preprocessed_text)
            progress.value = 100
            # Display predictions
            result = "<p style='color: green; font-size: 20px; background-color:
↪ #CCFFCC; padding: 10px; border-radius: 5px;'>GossipCop Prediction: Not␣
↪Fake</p>" if prediction[0][0] <= 0.5 else "<p style='color: red; font-size:␣
↪20px; background-color: #FFCCCC; padding: 10px; border-radius: 5px;
↪'>GossipCop Prediction: Fake</p>"

        else:
            # Preprocess the text
            preprocessed_text = preprocess_text(input_text,␣
↪tokenizer_politifact)
            progress.value = 90
            # Making predictions
            prediction = model_politifact(preprocessed_text)
            progress.value = 100
            # Display predictions
            result = "<p style='color: green; font-size: 20px; background-color:
↪ #CCFFCC; padding: 10px; border-radius: 5px;'>PoliticFact Prediction: Not␣
↪Fake</p>" if prediction[0][0] <= 0.5 else "<p style='color: red; font-size:␣
↪20px; background-color: #FFCCCC; padding: 10px; border-radius: 5px;
↪'>PoliticFact Prediction: Fake</p>"

        display(HTML(result))
        progress.value = 0

button.on_click(on_button_clicked)

# Clear button click event handler
```

```python
def on_clear_clicked(b):
    text_input.value = ''
    output.clear_output()

clear_button.on_click(on_clear_clicked)

# Display widgets
display(news_type, text_input, button, clear_button, output)
```

```
Requirement already satisfied: tensorflow==2.17.0 in
/usr/local/lib/python3.10/dist-packages (2.17.0)
Requirement already satisfied: gdown==5.1.0 in /usr/local/lib/python3.10/dist-
packages (5.1.0)
Requirement already satisfied: joblib==1.4.2 in /usr/local/lib/python3.10/dist-
packages (1.4.2)
Requirement already satisfied: ipywidgets==7.7.1 in
/usr/local/lib/python3.10/dist-packages (7.7.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (3.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (0.4.0)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (3.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (24.1)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (71.0.4)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (1.64.1)
Requirement already satisfied: tensorboard<2.18,>=2.17 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (2.17.0)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow==2.17.0) (3.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow==2.17.0) (1.26.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from gdown==5.1.0) (4.12.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from gdown==5.1.0) (3.15.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from gdown==5.1.0) (4.66.5)
Requirement already satisfied: ipykernel>=4.5.1 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets==7.7.1) (5.5.6)
Requirement already satisfied: ipython-genutils~=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets==7.7.1) (0.2.0)
Requirement already satisfied: traitlets>=4.3.1 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets==7.7.1) (5.7.1)
Requirement already satisfied: widgetsnbextension~=3.6.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets==7.7.1) (3.6.8)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.10/dist-
packages (from ipywidgets==7.7.1) (7.34.0)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets==7.7.1) (3.0.13)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from
astunparse>=1.6.0->tensorflow==2.17.0) (0.44.0)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-
packages (from ipykernel>=4.5.1->ipywidgets==7.7.1) (6.1.12)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-
packages (from ipykernel>=4.5.1->ipywidgets==7.7.1) (6.3.3)
Collecting jedi>=0.16 (from ipython>=4.0.0->ipywidgets==7.7.1)
  Using cached jedi-0.19.1-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets==7.7.1) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets==7.7.1) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets==7.7.1)
```

```
(3.0.47)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets==7.7.1) (2.16.1)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets==7.7.1) (0.2.0)
Requirement already satisfied: matplotlib-inline in
/usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets==7.7.1)
(0.1.7)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets==7.7.1) (4.9.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow==2.17.0) (13.8.0)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow==2.17.0) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow==2.17.0) (0.12.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.21.0->tensorflow==2.17.0) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.21.0->tensorflow==2.17.0) (3.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.21.0->tensorflow==2.17.0) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.21.0->tensorflow==2.17.0) (2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow==2.17.0) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow==2.17.0) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow==2.17.0) (3.0.4)
Requirement already satisfied: notebook>=4.4.1 in
/usr/local/lib/python3.10/dist-packages (from
widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (6.5.5)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->gdown==5.1.0) (2.6)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
/usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown==5.1.0)
(1.7.1)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from
jedi>=0.16->ipython>=4.0.0->ipywidgets==7.7.1) (0.8.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
```

```
(from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (3.1.4)
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.10/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(24.0.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(23.1.0)
Requirement already satisfied: jupyter-core>=4.6.1 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (5.7.2)
Requirement already satisfied: nbformat in /usr/local/lib/python3.10/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(5.10.4)
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.10/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(6.5.4)
Requirement already satisfied: nest-asyncio>=1.5 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (1.6.0)
Requirement already satisfied: Send2Trash>=1.8.0 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (0.18.1)
Requirement already satisfied: prometheus-client in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (0.20.0)
Requirement already satisfied: nbclassic>=0.4.7 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (1.1.0)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.10/dist-packages (from jupyter-
client->ipykernel>=4.5.1->ipywidgets==7.7.1) (2.8.2)
Requirement already satisfied: ptyprocess>=0.5 in
/usr/local/lib/python3.10/dist-packages (from
pexpect>4.3->ipython>=4.0.0->ipywidgets==7.7.1) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-
packages (from prompt-
toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=4.0.0->ipywidgets==7.7.1)
(0.2.13)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from
werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow==2.17.0) (2.1.5)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from
rich->keras>=3.2.0->tensorflow==2.17.0) (3.0.0)
Requirement already satisfied: platformdirs>=2.5 in
```

```
/usr/local/lib/python3.10/dist-packages (from jupyter-
core>=4.6.1->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(4.2.2)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow==2.17.0)
(0.1.2)
Requirement already satisfied: notebook-shim>=0.2.3 in
/usr/local/lib/python3.10/dist-packages (from
nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(0.2.4)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages
(from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(4.9.4)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-
packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.10/dist-packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(0.4)
Requirement already satisfied: jupyterlab-pygments in
/usr/local/lib/python3.10/dist-packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(0.3.0)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.10/dist-packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in
/usr/local/lib/python3.10/dist-packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(0.10.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(1.5.1)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-
packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(1.3.0)
Requirement already satisfied: fastjsonschema>=2.15 in
/usr/local/lib/python3.10/dist-packages (from
```

```
nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(2.20.0)
Requirement already satisfied: jsonschema>=2.6 in
/usr/local/lib/python3.10/dist-packages (from
nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(4.23.0)
Requirement already satisfied: argon2-cffi-bindings in
/usr/local/lib/python3.10/dist-packages (from
argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1)
(21.2.0)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1-
>widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat-
>notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat-
>notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1-
>widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (0.20.0)
Requirement already satisfied: jupyter-server<3,>=1.8 in
/usr/local/lib/python3.10/dist-packages (from notebook-shim>=0.2.3-
>nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets==7.7.1) (1.24.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-
packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1-
>widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (1.17.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets==7.7.1) (0.5.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1-
>widgetsnbextension~=3.6.0->ipywidgets==7.7.1) (2.22)
Requirement already satisfied: anyio<4,>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server<3,>=1.8->notebook-s
him>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets==7.7.1) (3.7.1)
Requirement already satisfied: websocket-client in
/usr/local/lib/python3.10/dist-packages (from jupyter-server<3,>=1.8->notebook-s
him>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets==7.7.1) (1.8.0)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3-
>nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets==7.7.1) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-
```

```
packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3-
>nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets==7.7.1) (1.2.2)
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Installing collected packages: jedi
Successfully installed jedi-0.19.1
```

```
Downloading…
From: https://drive.google.com/uc?id=1tgL8ITlc61l0BnjDLbnyOJ8zghtNDjVL
To: /content/best_cnn_model_gossipcop.keras
100%|      | 2.68M/2.68M [00:00<00:00, 167MB/s]
Downloading…
From: https://drive.google.com/uc?id=12e6MIz9qKCbLu6cz7zyRytOVAwKPOMYI
To: /content/tokenizer_gossipcop.pkl
100%|      | 479k/479k [00:00<00:00, 50.8MB/s]
Downloading…
From: https://drive.google.com/uc?id=1PT-7nFgXTOqISSRqEe9jTMJdAro5aQ3j
To: /content/label_encoder_gossipcop.pkl
100%|      | 343/343 [00:00<00:00, 913kB/s]
Downloading…
From: https://drive.google.com/uc?id=14wU7pqr6tlpg3bE89lx3Y-NfOO_Q4Mwy
To: /content/best_cnn_model_politifact.keras
100%|      | 9.24M/9.24M [00:00<00:00, 228MB/s]
Downloading…
From: https://drive.google.com/uc?id=19nmqTyZoXZcbc24Gw7lYEKyLNVoKGine
To: /content/tokenizer_politifact.pkl
100%|      | 104k/104k [00:00<00:00, 39.9MB/s]
Downloading…
From: https://drive.google.com/uc?id=1wHBlUc4xLZYdbkW8yVmmVQtxpKyHAftD
To: /content/label_encoder_politifact.pkl
100%|      | 343/343 [00:00<00:00, 882kB/s]
```

```
Setup complete!
```

```
Dropdown(description='News Type:', options=('Gossip', 'Political'),␣
 ↪value='Gossip')
```

```
Textarea(value='', description='News Title:', layout=Layout(height='200px',␣
 ↪width='100%'), placeholder='Type t…
```

```
Button(description='Check Fake News', style=ButtonStyle(), tooltip='Click to␣
 ↪check if the news is fake or not'…
```

```
Button(description='Clear', style=ButtonStyle(), tooltip='Clear the input text')
```

```
Output()
```

[ ]:

```
# This cell will be removed after successful conversion of notebook to PDF, to␣
 ↪not depend on connection to Google Drive
```

```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Install LaTeX packages necessary for converting notebooks to PDF
!apt-get update
!apt-get install -y texlive-xetex texlive-fonts-recommended␣
 ↪texlive-plain-generic texlive-latex-extra pandoc

# Convert the notebook to PDF
!jupyter nbconvert --to pdf "/content/drive/My Drive/Colab Notebooks/
 ↪FakeNewsNetInference.ipynb"
```

```
Mounted at /content/drive
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,626 B]
Get:5 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease [1,581 B]
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:7 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Get:8 https://r2u.stat.illinois.edu/ubuntu jammy Release [5,713 B]
Get:9 https://r2u.stat.illinois.edu/ubuntu jammy Release.gpg [793 B]
Get:10 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
[18.1 kB]
Hit:11 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:13
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
Packages [976 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,499
kB]
Get:15 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,293 kB]
Get:16 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy/main amd64
Packages [27.8 kB]
Get:17 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,575 kB]
Fetched 14.7 MB in 3s (4,906 kB/s)
Reading package lists… Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
```

```
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc-data
poppler-data
  preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick ruby-
xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base texlive-
binaries
  texlive-latex-base texlive-latex-recommended texlive-pictures tipa xfonts-
encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java texlive-
luatex
  pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc nodejs php python
libjs-mathjax
  libjs-katex citation-style-language-styles poppler-utils ghostscript fonts-
japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-
arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv | postscript-
viewer perl-tk xpdf
  | pdf-viewer xzdec texlive-fonts-recommended-doc texlive-latex-base-doc
python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-
extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex texlive-
pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc
pandoc-data
  poppler-data preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-
```

```
webrick ruby-xmlrpc
  ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
texlive-binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-extra texlive-
latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa xfonts-encodings
xfonts-utils
0 upgraded, 58 newly installed, 0 to remove and 49 not upgraded.
Need to get 202 MB of archives.
After this operation, 728 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.9 [752 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.9 [5,033 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-
gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-gfm-
extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]
```

```
Get:20 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.7 [50.1 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
Get:28 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ruby-webrick all
1.7.0-3 [51.8 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.7 [5,113 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc-data all
2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc amd64
2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
```

```
Get:44 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 4s (51.1 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 123597 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
```

```
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.9_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.9) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.9_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.9) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack …/17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack …/18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcommons-parent-java.
```

```
Preparing to unpack …/19-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/20-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/21-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/23-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/24-ruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/25-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/26-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/27-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/28-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/29-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/31-libruby3.0_3.0.2-7ubuntu2.7_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/32-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/33-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
```

```
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/36-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/38-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/39-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package pandoc-data.
Preparing to unpack …/40-pandoc-data_2.9.2.1-3ubuntu2_all.deb …
Unpacking pandoc-data (2.9.2.1-3ubuntu2) …
Selecting previously unselected package pandoc.
Preparing to unpack …/41-pandoc_2.9.2.1-3ubuntu2_amd64.deb …
Unpacking pandoc (2.9.2.1-3ubuntu2) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/42-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/43-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/44-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/45-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/47-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/48-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/49-texlive-latex-base_2021.20220204-1_all.deb …
```

```
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/50-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/51-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/52-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/53-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/54-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/55-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/56-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/57-texlive-xetex_2021.20220204-1_all.deb …
Unpacking texlive-xetex (2021.20220204-1) …
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
Setting up tex-common (6.17) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
```

```
Setting up ruby-webrick (1.7.0-3) …
Setting up libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up pandoc-data (2.9.2.1-3ubuntu2) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.9) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.9) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up pandoc (2.9.2.1-3ubuntu2) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
Setting up tipa (2:1.3-21) …
```

```
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.7) …
Setting up ruby3.0 (3.0.2-7ubuntu2.7) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.4) …
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a
symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

Processing triggers for tex-common (6.17) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time…
```