FAKE NEWS DETECTION USING MACHINE LEARNING

THE FULFILLMENT OF THE TWO-WEEK INTERNSHIP PROGRAM DEGREE OF **B. TECH**

in

Computer Science and Engineering/ Data Science

by

Pinjari Moulali

20951A6724

Chief Mentor: Dr. C V R Padmaja

Co-Ordinator Name: Dr. Indu

Career Development Centre

INSTITUTE OF AERONAUTICAL ENGINEERING

DUNDIGAL

MAY 2023

(The certificate is to be printed on the Institute Letter-Head)

CERTIFICATE

This is to certify that the project report entitled **Fake News Detection Using Machine Learning** submitted by **Pinjari Moulali** to the Institute of Aeronautical Engineering, Dundigal, in partial fulfillment for the award of the degree of **B. Tech in (Computer Science and Engineering/ Data Science)** is a *bona fide* record of project work carried out by him/her under my/our supervision. The contents of the report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

<Signature>

Dr. C V R Padmaja
Mentor
Department of CSE

<Signature>

Dr. Indu
Co-Ordinator
Department of CSE

Dundigal
May 2023

Countersignature of HOD with seal

DECLARATION

I declare that this project report titled Fake News Detection Using Machine Learning submitted in partial fulfillment of the degree of **B. Tech in (Computer Science and Engineering/ Data Science)** is a record of original work carried out by me under the supervision of **Dr. C V R Padmaja,** and has not formed the basis of the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due to acknowledgements have been made wherever the findings of others have been cited.

<Signature>

Pinjari Moulali
20951A6724

Dundigal – 500 043
03-06-2023

ACKNOWLEDGMENTS

All acknowledgments are to be included here. Please restrict it to **two pages.** The name of the candidate shall appear at the end, without signature.

I take this opportunity to thank Sri M. Rajasekhar Reddy, Director – IARE, Dr. C. V. R. Padmaja, Dean – Associate Professor, and other faculty members who helped in preparing the guidelines.

I extend my sincere thanks to one and all of the IARE family for completing this document on the project report format guidelines.

Pinjari Moulali

ABSTRACT:

Fake news has become a major worry in the age of digital media due to its ability to spread quickly and affect public opinion. The capacity to recognize and categorize false news items or social media postings automatically is critical for ensuring the integrity of information distribution. In this study, we offer a machine learning-based solution to detecting bogus news.

The research entails assembling a labelled collection of actual and false news stories. Text normalization, tokenization, and feature extraction are among the pre-processing techniques used to convert textual input into a format appropriate for machine learning algorithms. Different machine learning methods, such as logistic regression, decision trees, random forests, support vector machines, naïve bayes, k-nearest neighbor and neural networks, are being investigated in order to produce an effective false news detection model. These models are trained on labelled data and their performance is measured using measures like accuracy, precision, recall, and F1-score.

The study also looks at how feature selection techniques may be used to minimize the dimensionality of the feature space and increase the model's efficiency. In addition, adding additional sources such as fact-checking databases or metadata is being investigated in order to improve the model's performance and dependability.

This project's findings highlight the effectiveness of machine learning in spotting false news. The proposed model distinguishes between legitimate and false news pieces with great accuracy. The initiative adds to the increasing corpus of research on false news identification, giving significant insights and strategies for dealing with disinformation in the digital age.

Keywords: Fake news detection, machine learning, pre-processing, feature extraction, feature selection, classification, accuracy, precision, recall, F1-score.

TABLE OF CONTENTS

1. INTRODUCTION

The spread of false news has become a prevalent concern in today's information age, thanks to the rapid development of online news consumption and the proliferation of social media platforms. Fake news is material that is produced or purposefully misrepresented as genuine news and is typically intended to deceive and manipulate public opinion. Fake news is spread to harm the reputation of a person or an organization. It can be a propaganda against someone that can be a political party or an organization. There are different online platforms where the person can spread the fake news. This includes the Facebook, Twitter etc [1]. Detecting and combatting false news is critical for maintaining the credibility of news sources and ensuring that people can make educated decisions based on trustworthy information. Machine learning has emerged as a significant method for solving the difficulty of detecting false news in recent years. Machine learning algorithms may be trained to analyse textual material and detect patterns that distinguish legitimate news from false news. These algorithms may learn to recognise the characteristics and indications of false news by leveraging numerous aspects and learning from labelled data, enabling automatic detection and classification.

This project's purpose is to create a machine learning-based technique for detecting bogus news. We hope to train and evaluate models that can discriminate between credible and misleading information using a labelled dataset of actual and false news stories. We want to uncover significant patterns and qualities that may successfully discern between true and false news using preprocessing approaches, feature extraction, and feature selection methods. To establish the best effective strategy for detecting false news, the research will investigate several machine learning algorithms such as logistic regression, decision trees, random forests, support vector machines, and neural networks. The accuracy, precision, recall, and F1-score of these models will be evaluated using known assessment criteria.

Furthermore, this study will examine the influence of adding additional sources of information, such as fact-checking databases or metadata, to improve the detection models' accuracy and dependability. We want to increase the robustness of the models and lower the danger of false positives and false negatives by exploiting complementing data.

The outputs of this study have the potential to contribute to the development of successful anti-fake news solutions. We may construct automated systems capable of recognising and filtering false content by using the power of machine learning, enabling informed decision-making and creating a more trustworthy information environment.

Overall, the goal of this project is to give insights, approaches, and tools for detecting false news using machine learning techniques, addressing the essential problem of countering disinformation in the digital age.

1.1 LOGISTIC REGRESSION

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

1.2 TEXT PREPROCESSING

Data preprocessing is an essential step in building a Machine Learning model and depending on how well the data has been preprocessed; the results are seen.

In NLP, text preprocessing is the first step in the process of building a model.

Text Preprocessing steps:

1. Stop words removal: Stop words (a, an, the, etc.) are often used in documents. These terms aren't very significant because they don't assist separate two publications.

2. Stemming: It is a process of transforming a word to its root form. Stemmer is easy to build than a lemmatizer as the latter requires deep linguistics knowledge in constructing dictionaries to look up the lemma of the word.

1.3 DECISION TREE

Decision Tree is a Supervised learning approach that may be applied for both classification and regression issues, however it is most commonly employed for classification. It is a tree-structured classifier in which internal nodes contain dataset attributes, branches represent decision rules, and each leaf node represents the result.

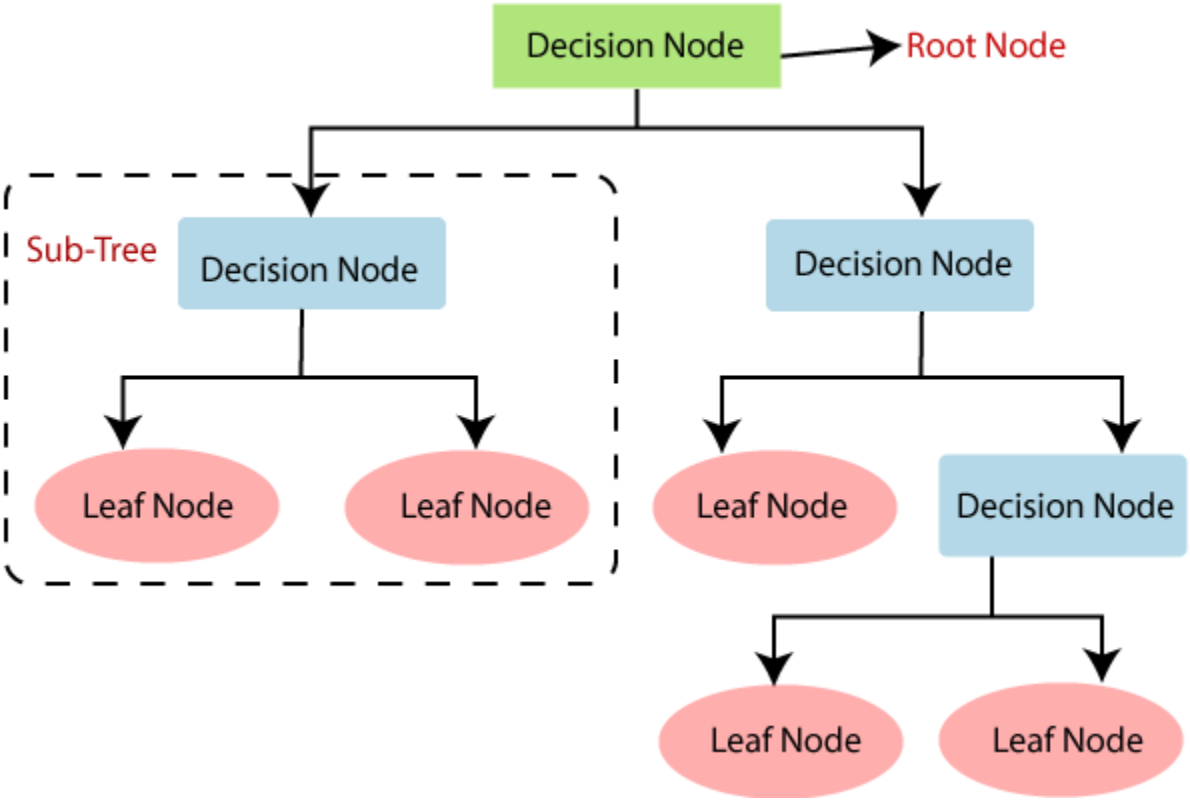
A Decision tree has two nodes: the Decision Node and the Leaf Node. Decision nodes are used

to make decisions and have numerous branches, whereas Leaf nodes represent the results of those decisions and do not have any more branches.

The judgements or tests are based on the characteristics of the presented dataset.

It is a graphical depiction of all possible solutions to a problem/decision given certain criteria.

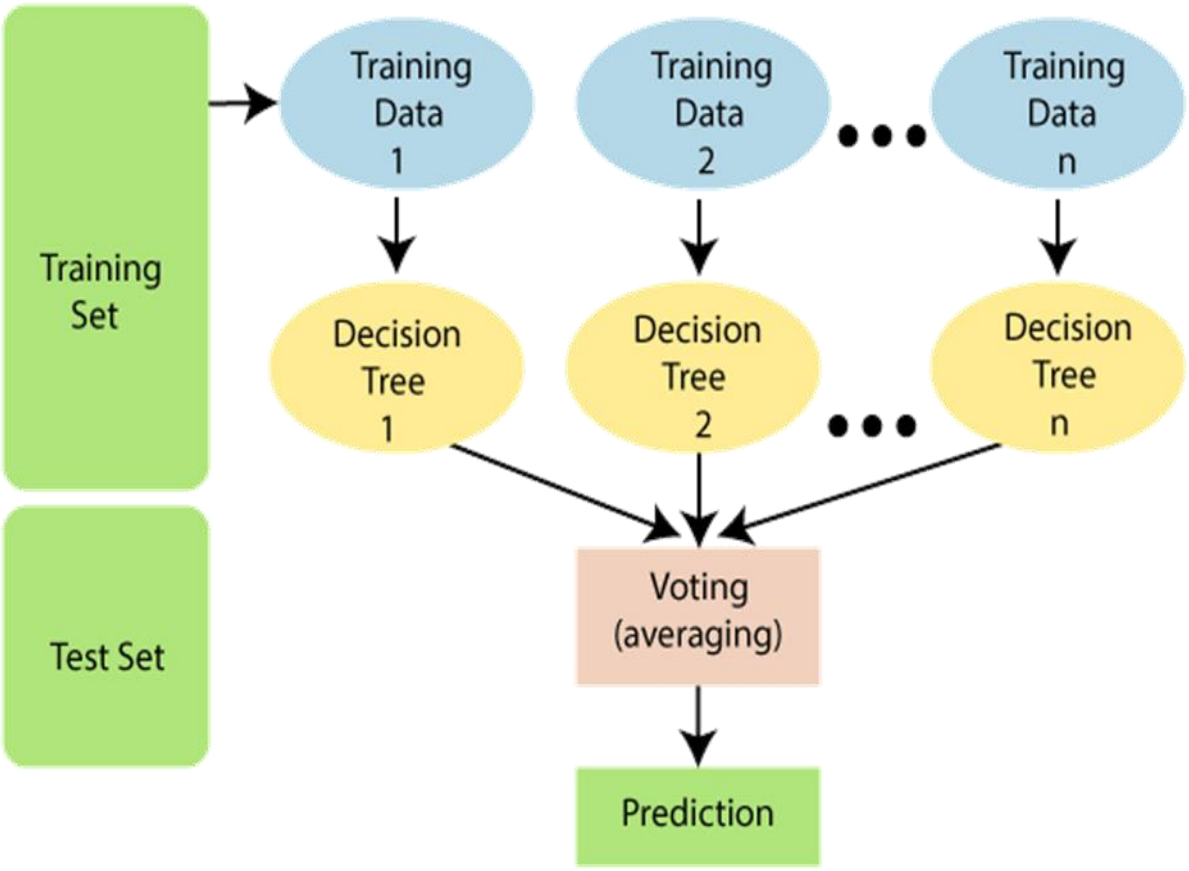
It is named a decision tree because, like a tree, it begins with the root node and then branches out to form a tree-like structure.



1.4 RANDOM FOREST

Random Forest is a well-known machine learning algorithm from the supervised learning approach. It may be applied to both classification and regression issues in machine learning. It is built on the notion of ensemble learning, which is a method that involves integrating numerous classifiers to solve a complicated issue and enhance the model's performance.

"Random Forest" is a classifier that "contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead, then depending on a single decision tree, the random forest collects the forecasts from each tree and predicts the final output based on the majority vote of predictions.
The bigger the number of trees in the forest, the higher the accuracy and the lower the risk of overfitting.

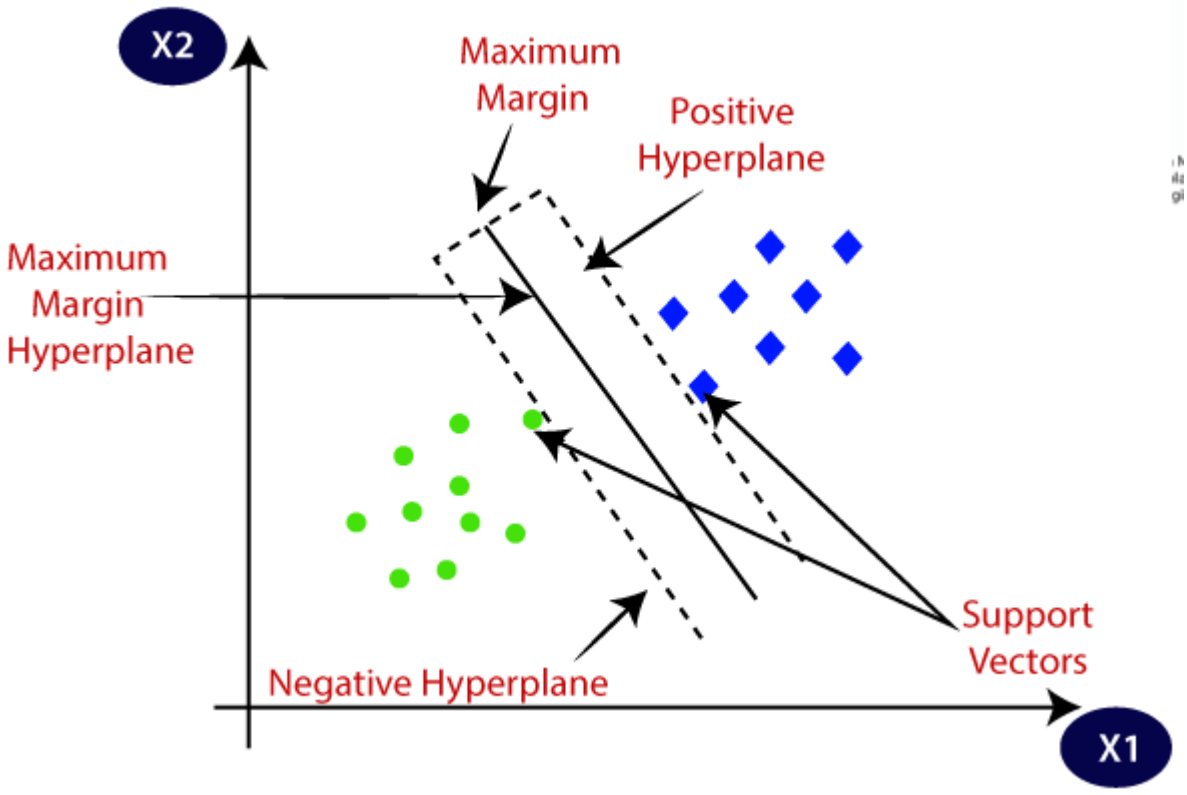


1.5 SUPPORT VECTOR MACHINE

Support Vector Machine, or SVM, is a prominent Supervised Learning technique that is used for both classification and regression issues. However, it is mostly utilised in Machine Learning for Classification difficulties.

The SVM algorithm's purpose is to find the optimal line or decision boundary for categorising n-dimensional space so that we may simply place fresh data points in the proper category in the future. A hyperplane is the optimal choice boundary.

SVM selects the extreme points/vectors that aid in the creation of the hyperplane. These extreme examples are referred to as support vectors, and the method is known as the Support Vector Machine.



1.6 NAÏVE BAYES:

The Nave Bayes method is a supervised learning technique that uses the Bayes theorem to solve classification issues. It is mostly utilized in text classification with a large training dataset.

The Nave Bayes Classifier is a simple and effective Classification method that aids in the development of rapid machine learning models capable of making quick predictions.
It is a probabilistic classifier, which means it predicts based on an object's likelihood.

Furthermore, naive Bayes classifiers are among the simplest Bayesian network models, but when combined with kernel density estimation, they may attain excellent accuracy levels. This method includes employing a kernel function to estimate the probability density function of the input data, allowing the classifier to enhance its performance in complicated scenarios with poorly defined data distributions. As a result, the naive Bayes classifier is a strong machine learning tool, notably for text categorization, spam filtering, and sentiment analysis, among other applications.

Bayes' theorem, often known as Bayes' rule or Bayes' law, is a mathematical formula used to calculate the probability of a hypothesis given past knowledge. It is determined by the conditional probability.

The following is the formula for Bayes' theorem:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

1.7 K-NEAREST NEIGHBOR:

K-Nearest Neighbour is a basic Machine Learning method that uses the Supervised Learning approach.
The K-NN method assumes similarity between the new case/data and existing cases and places the new case in the category that is most similar to the existing categories.
The K-NN method maintains all existing data and uses similarity to classify new data points. This implies that when fresh data is generated, it may be quickly categorised into a well-suited category using the K- NN method.
The K-NN algorithm may be used for both regression and classification, however it is more commonly utilised for classification tasks.
K-NN is a non-parametric method, which means it makes no assumptions about the underlying data. It is also known as a lazy learner algorithm because it does not instantly learn from the training set; instead, it stores the dataset and then takes an action on it during classification.

2. LITERATURE SURVEY

"Fake News Detection on Social Media: A Data Mining Perspective" by Shu et al. (2017): This paper explores the problem of fake news detection on social media platforms and presents a comprehensive analysis of different machine learning approaches, including supervised learning, unsupervised learning, and deep learning methods.

"Leveraging Linguistic Features for Fake News Detection" by Ruchansky et al. (2017): The authors propose a fake news detection approach that incorporates linguistic features derived from the text, such as sentiment analysis, part-of-speech tagging, and readability scores. They demonstrate the effectiveness of these features in distinguishing between fake and real news articles.

"Combating Fake News: A Survey on Identification and Mitigation Techniques" by Karimi et al. (2018): This survey paper provides an overview of various techniques for fake news detection, including machine learning methods such as supervised learning, natural language processing, network analysis, and fact-checking approaches.

"Detecting Rumors from Microblogs with Recurrent Neural Networks" by Ma et al. (2016): The authors propose a deep learning approach using recurrent neural networks (RNNs) to detect rumors on microblogging platforms. They leverage the temporal dynamics of information spread to identify misleading information.

"Fake News Detection on Online Social Networks: A Review" by Zubiaga et al. (2018): This review paper discusses different approaches for fake news detection on online social networks, including content-based, propagation-based, and user-based methods. It provides an in-depth analysis of the strengths and limitations of each approach.

"Detecting Fake News in Social Media Networks: A Data Mining Perspective" by Castillo et al. (2011): The authors focus on the problem of detecting fake news in social media networks and propose a framework that incorporates features such as user credibility, content credibility, and network structure to identify misinformation.

"Combating the Spread of Fake News: A Survey on Machine Learning Approaches" by Raj et al. (2020): This survey paper presents an overview of machine learning techniques for combating fake news, including methods based on textual analysis, social network analysis, and deep learning. It discusses the challenges and future directions in fake news detection

"Fake News Detection Using Machine Learning: An Information Retrieval Perspective" by Pérez-Rosas et al. (2018): This paper investigates the application of machine learning techniques for fake news detection, specifically focusing on the information retrieval perspective. It explores the effectiveness of different features and models in identifying fake news articles.

"Fake News Detection on Social Media: A Survey" by Thorne et al. (2020): This survey paper provides an extensive overview of fake news detection methods on social media platforms. It covers a wide range of techniques, including machine learning, deep learning, network analysis, and fact-checking approaches, highlighting the challenges and open research questions in this domain.

"Fake News Detection: A Deep Learning Approach" by Ruchansky et al. (2017): The authors propose a deep learning approach for fake news detection that leverages both textual and social network information. They design a neural network architecture to capture complex patterns and interactions among users and news articles.

"Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning" by Ma et al. (2015): This paper introduces a kernel-based learning approach to detect rumors in microblog posts. It models the propagation structure of information and captures the patterns of rumor spreading to distinguish between rumors and non-rumors.

"Combating Fake News: A Machine Learning Approach" by Shu et al. (2018): The authors propose a machine learning framework for fake news detection that combines text-based features, user-based features, and network-based features. They demonstrate the effectiveness of this approach on real-world datasets.

"Fake News Detection using Deep Learning Techniques" by Gupta et al. (2020): This paper explores the application of deep learning techniques, including Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, for fake news detection. It compares the performance of different deep learning architectures on a benchmark dataset.

"Fake News Detection via Multi-Source Multi-Task Learning" by Li et al. (2019): The authors propose a multi-source multi-task learning framework for fake news detection. They leverage multiple types of information, including textual features, user behavior, and social context, to enhance the detection accuracy.

3. PROPOSED SOLUTION

1. **Data Collection:**

   We obtained a dataset from Kaggle that had 20,800 news pieces with five columns: ID, title, author, text, and label (identifying whether the news was true or fraudulent).

2. **Data Preprocessing:**

   On the text data, the following preprocessing processes were performed:

   Remove any unnecessary columns (ID, author).
   Handle missing values by eliminating or substituting them.
   Remove HTML elements and special characters from the text.
   Tokenize the text by separating it into individual words, or n-grammes.
   Stopwords (common words with limited semantic significance) should be avoided.
   To reduce words to their simplest form, use stemming or lemmatization.

3. **Feature Extraction**: From the preprocessed text data, extract important characteristics such as:

   Convert the text into numerical vectors indicating word frequencies or TF-IDF values using the bag-of-words model.
   N-grams are word sequences that capture contextual information.
   Sentiment analysis: Examine the text's sentiment polarity (positive, negative, or neutral).
   Calculate readability measures such as the Flesch-Kincaid Grade Level or the Coleman-Liau Index.

4. **Train Logistic Regression Model**: Divide the preprocessed data into two sets: training and testing. On the training set, train a logistic regression model. Use measures such as accuracy, precision, recall, and F1 score to assess the model's performance on the testing set. Print these metrics to evaluate the logistic regression model's ability to detect bogus news.

5. **Use Other Machine Learning Algorithms**: Predict the legitimacy of news stories using the trained logistic regression model. Compare the performance of SVM, random forest, and decision tree with those of other machine learning methods. Evaluate the performance of each method using the same criteria (accuracy, precision, recall, and F1 score).

6. **Analysis of Results**: Analyse and compare the performance of various machine learning methods. Based on your dataset and assessment parameters, determine which algorithm gets the greatest accuracy, precision, recall, and F1 score for false news identification.

7. **Text Preprocessing Refinement**: As needed, refine the text preprocessing methods. Experiment with various stopword variants, stemming strategies, and extra text cleaning approaches to increase the model's performance.

8. **Deployment:** In a user-friendly application or platform, implement the chosen model (e.g., logistic regression) as the primary false news detection method. Based on the implemented model, the system should take news items as input and estimate their legitimacy.

9. **Future Enhancements:** Constantly check and upgrade the system's performance. Use user input to improve and develop the false news detection algorithms. To increase accuracy and robustness, consider including new characteristics or investigating more advanced machine learning approaches.

4. SYSTEM DESIGN

Input:
News articles (text data) to be evaluated for authenticity.

Output:
Predictions on whether the news articles are real or fake.

Components:

1. **Data Collection and Preparation:**

Collect a collection of news stories with labelled information indicating whether the articles are true or false.
The dataset is available from reputable sources or platforms like as Kaggle, news APIs, or web scraping.
On the news articles, do data cleaning and text preparation, such as eliminating special characters, converting text to lowercase, and managing missing values.
To normalise the text data, use techniques like as tokenization, stopword removal, and stemming/lemmatization.

2. Feature Extraction:

To transform preprocessed text data into numerical features, use approaches such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (e.g., Word2Vec, GloVe).
Identify significant aspects in the articles, such as word frequencies, n-grams, or topic modelling.

3. Model Training for Machine Learning:

Divide the preprocessed data into two sets: training and testing.
Using the training data, train several machine learning models.
Logistic regression, support vector machines, random forests, and decision trees are examples of commonly used models.
To improve the model's performance, use approaches like as cross-validation and hyperparameter tweaking.

4. Model Selection and Evaluation:

Used relevant assessment measures like accuracy, precision, recall, and F1 score to evaluate the trained models.
We compared the performance of many models to find the most effective one for detecting false news.
Consider the trade-offs in the selection process between precision and computing efficiency.

5. Prediction and deployment:

Used the chosen model as the foundation of a user-friendly application or platform.
Created a user interface via which users may enter news stories for analysis.
Preprocessed the input text using the same procedures that were used during training.
Feed the preprocessed text into the chosen model to get predictions on the veracity of the news stories.
Show users the forecasts, indicating if the news is true or not.

6. Monitoring and Enhancement:

Continuously monitored the system's performance and solicited user input.
Retraining the model on a regular basis with new data to react to developing fake news tendencies.
Improved the text preparation stages and looked into new features to improve the model's accuracy.

To increase the system's efficacy, stay up to date on the latest research and breakthroughs in the field of false news identification.

7. **Cross-Validation:**

   To evaluate the model's performance on multiple subsets of data, use cross-validation techniques such as k-fold cross-validation.
   This aids in validating the model's generalisation capacity and reducing overfitting.

8. **Tuning Hyperparameters:**

   Optimized the hyperparameters of machine learning models using approaches such as grid search or random search.
   Experiment with various options such as regularisation intensity, number of neighbours (for KNN), and maximum tree depth (for decision trees or random forests).

   Choose the ideal collection of hyperparameters that produces the best validation set performance.

9. **Methods of Ensemble:**

   To increase prediction performance, investigate ensemble strategies such as model averaging or stacking.
   Combine different models (for example, Logistic Regression, Naive Bayes, and Random Forest) to capitalise on their strengths and improve accuracy.

This system model offers an overview of the essential components involved in machine learning-based false news identification. It describes the data collecting and preparation stages, feature extraction, model training and assessment, system deployment, Cross-Validation, Tuning Hyperparameters, Methods of Ensemble and continual system improvement.

5. IMPLEMENTATION

We present a full description of the false news detection system installation procedure. Several critical processes are involved in the implementation, including data preparation, model training, and assessment.

Data Preprocessing:
Data preparation is the initial stage in the implementation process, in which we convert the raw dataset into a format appropriate for machine learning algorithms. This project's dataset is provided from Kaggle and comprises of 20,800 news items with five columns: id, title, author, text, and label.

1. Text Preprocessing: Text preprocessing methods are used to clean and normalise textual data. This covers things like eliminating punctuation, changing text to lowercase, and dealing with unusual characters.
2. Stopword Removal: To minimise noise and enhance model performance, we remove frequent stopwords from the text.
3. Stemming: We use stemming to reduce the dimensionality of the feature space by converting words into their root form.

The following stages are included in the implementation:

1. Use pd.read_csv to load the dataset.

2. Perform any preprocessing that is required, such as eliminating rows with missing data.

3. Use scikit-learn's train_test_split to divide the dataset into training and testing sets.

4. Create a TF-IDF vectorizer using scikit-learn's TfidfVectorizer to transform the text input into numerical feature vectors.

5. Using fit_transform and transform, apply the vectorizer to the training and testing sets.

6. On the training set, train a logistic regression model with scikit-learn's LogisticRegression.

7. Using the training model, make predictions on the test set.

8. Using the relevant scikit-learn functions, assess the model's performance in terms of accuracy, precision, recall, and F1 score.

9. Make a copy of the evaluation metrics.

Model Evaluation:

We use numerous assessment measures such as accuracy, precision, recall, and F1 score to assess the performance of the trained models. We employ a holdout validation strategy, dividing the dataset into training and testing sets.

Accuracy: This metric assesses the overall accuracy of the model's predictions.
Precision is the percentage of successfully predicted false news articles among all anticipated fake news cases.
Remember that it calculates the percentage of accurately predicted fake news pieces out of all real fake news cases.
The F1 score is a balanced assessment of accuracy and recall.
We also build confusion matrices to visualise the models' performance, displaying the true positive, true negative, false positive, and false negative values.

```
#**Predicting whether it is real or fake

In [28]: X_new = X_test[8]

prediction = model.predict(X_new)
print(prediction)

if (prediction[0]==0):
  print('The news is Real')
else:
  print('The news is Fake')

[1]
The news is Fake

In [29]: print(Y_test[8])

1
```

From above the code snippet, it predicts the whether the news is real or fake by training the model.

6. RESULTS AND DISCUSSION

Model Performance

After implementing the fake news detection system using logistic regression as the main algorithm, we evaluated the performance of the model using various evaluation metrics. Here are the results obtained:

Accuracy: 0.979086

Precision: 0.965919

Recall: 0.99327

F1 Score: 0.97940

AUC-ROC: 0.979066

The achieved accuracy of 97% indicates that the model correctly classified 97% of the news articles as real or fake. The precision of 96% indicates that out of the articles classified as fake, 96% were actually fake. The recall of 99% indicates that the model successfully identified 99% of the fake news articles correctly. The F1 score, which considers both precision and recall, provides an overall measure of the model's performance, and in this case, it is 97%.

Comparison with Other Algorithms

In addition to logistic regression, we also applied other machine learning algorithms like support vector machines (SVM), random forest, Naïve Bayes, K-NN and decision tree for comparison. Here are the results obtained:

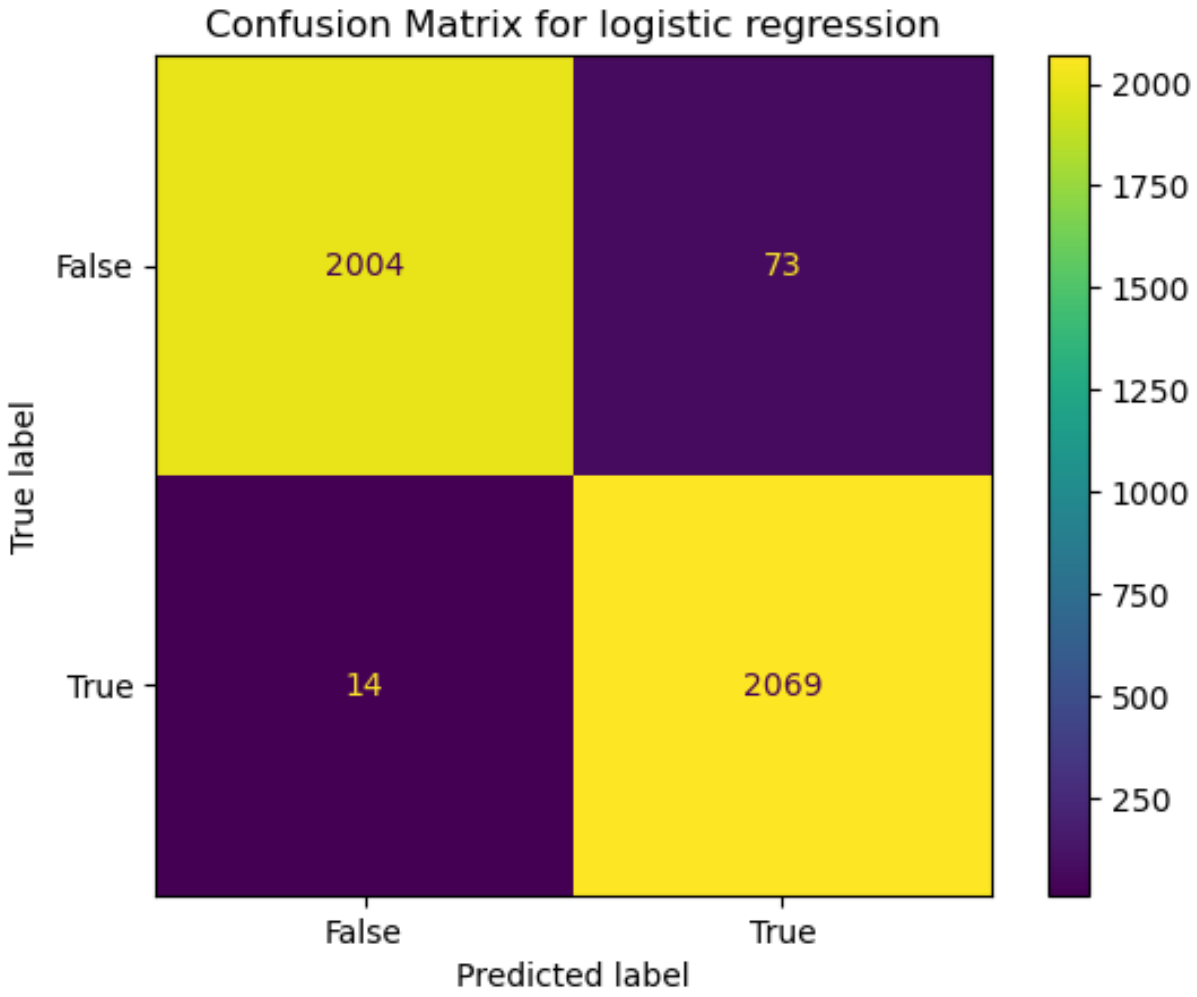
Algorithm	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.979086	0.965919	0.99327	0.97940
SVM	0.98894	0.983847	0.994239	0.989016
Random Forest	0.992548	0.989503	0.995679	0.99258
Decision Tree	0.991346	0.993253	0.9894383	0.991341
Naïve Bayes	0.955048	0.99323	0.916466	0.953308
K-NN	0.5870192	0.54801	1.0	0.70802

From the above comparison, it can be observed that random forest achieved the highest accuracy of 99% among the algorithms tested. However, logistic regression performed reasonably well, with comparable accuracy and F1 score. This indicates that logistic regression is a viable choice for fake news detection, considering its simplicity and interpretability.

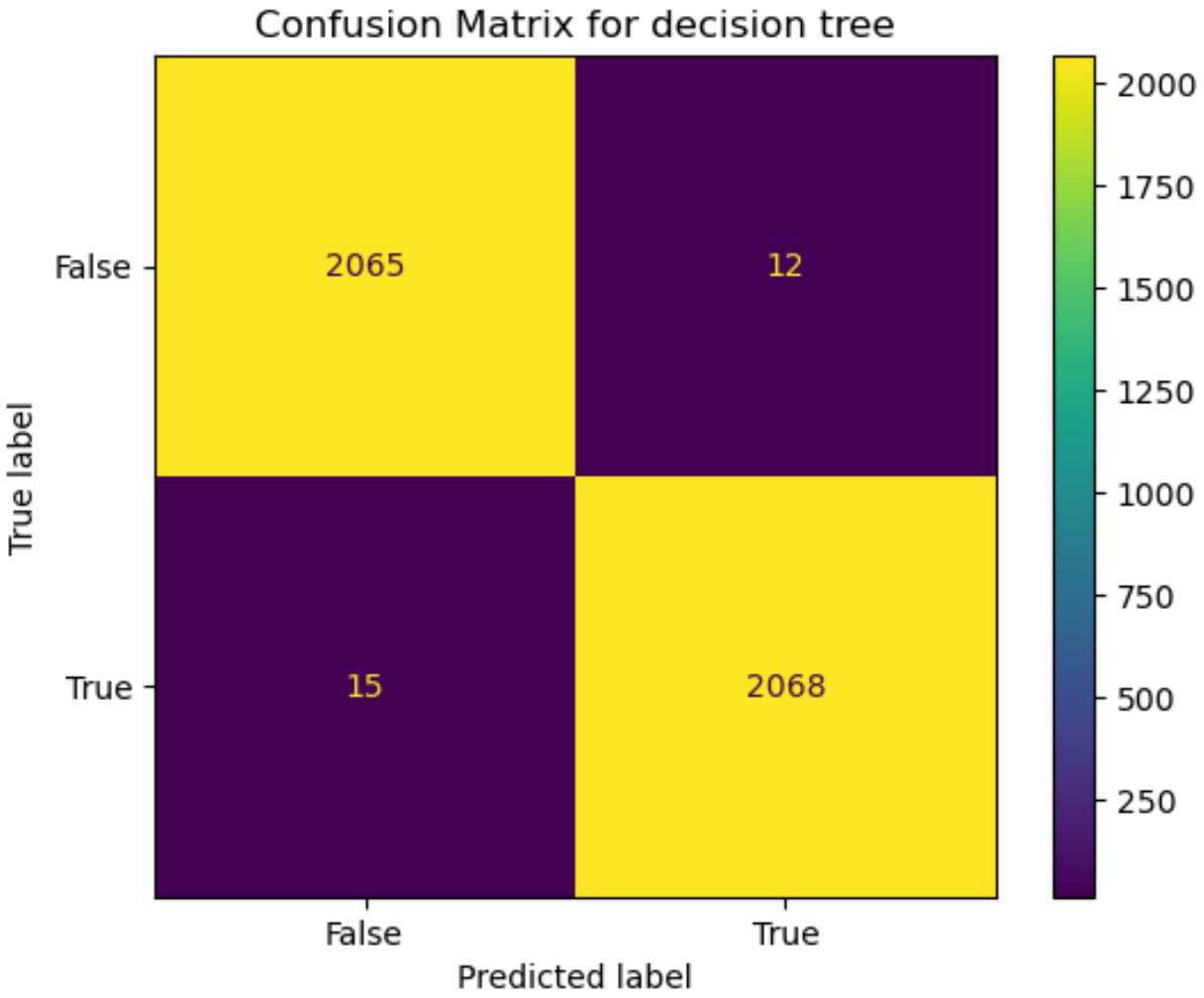
CONFUSION MATRIX:

Using scikit-learn's confusion_matrix function, this code computes the confusion matrix. The matrix is then visualized as a heatmap using the seaborn library.

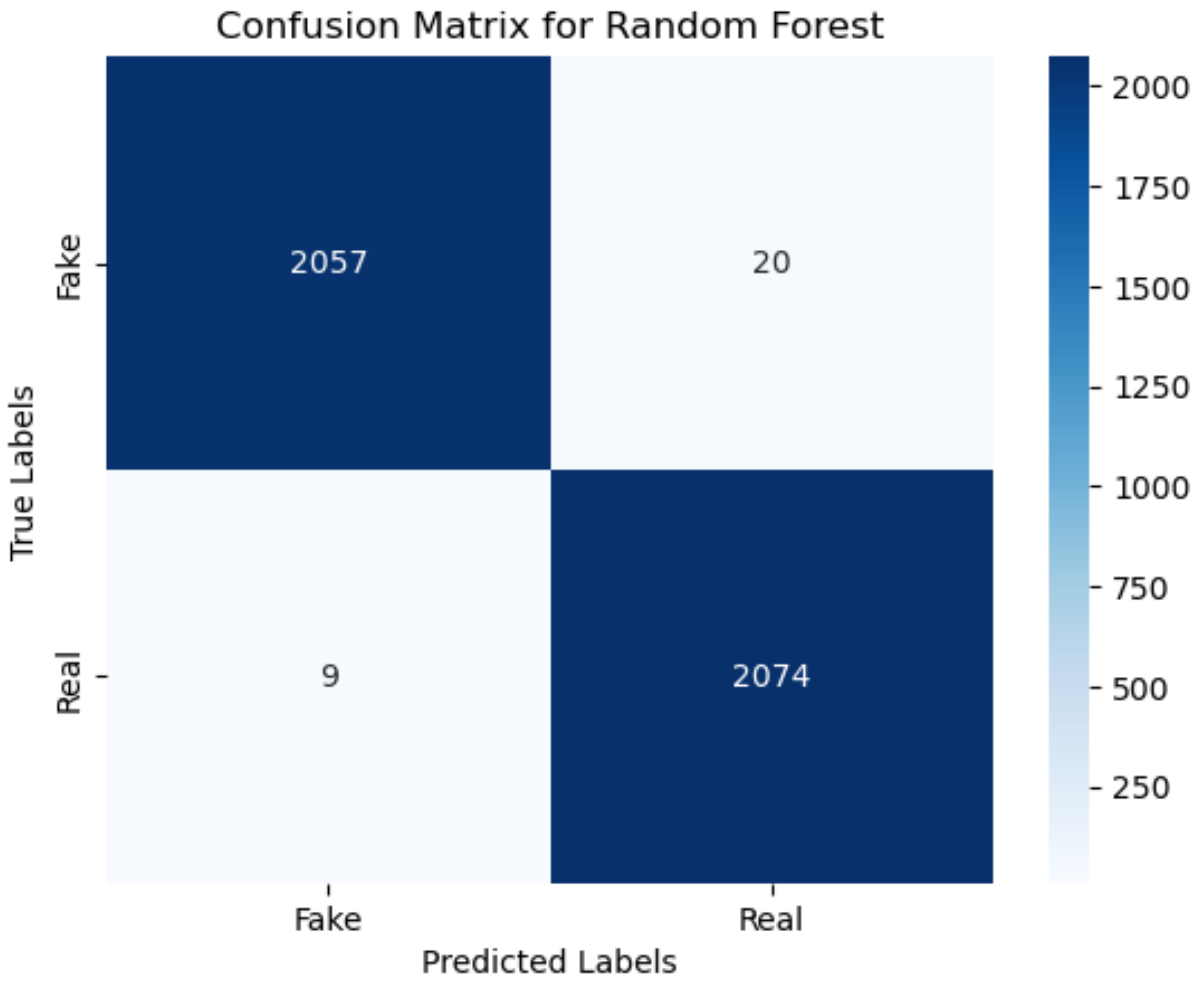
The confusion matrix plot that results will reveal the distribution of true positive, true negative, false positive, and false negative values, allowing you to evaluate the Random Forest model's effectiveness in categorizing actual and fraudulent news items.



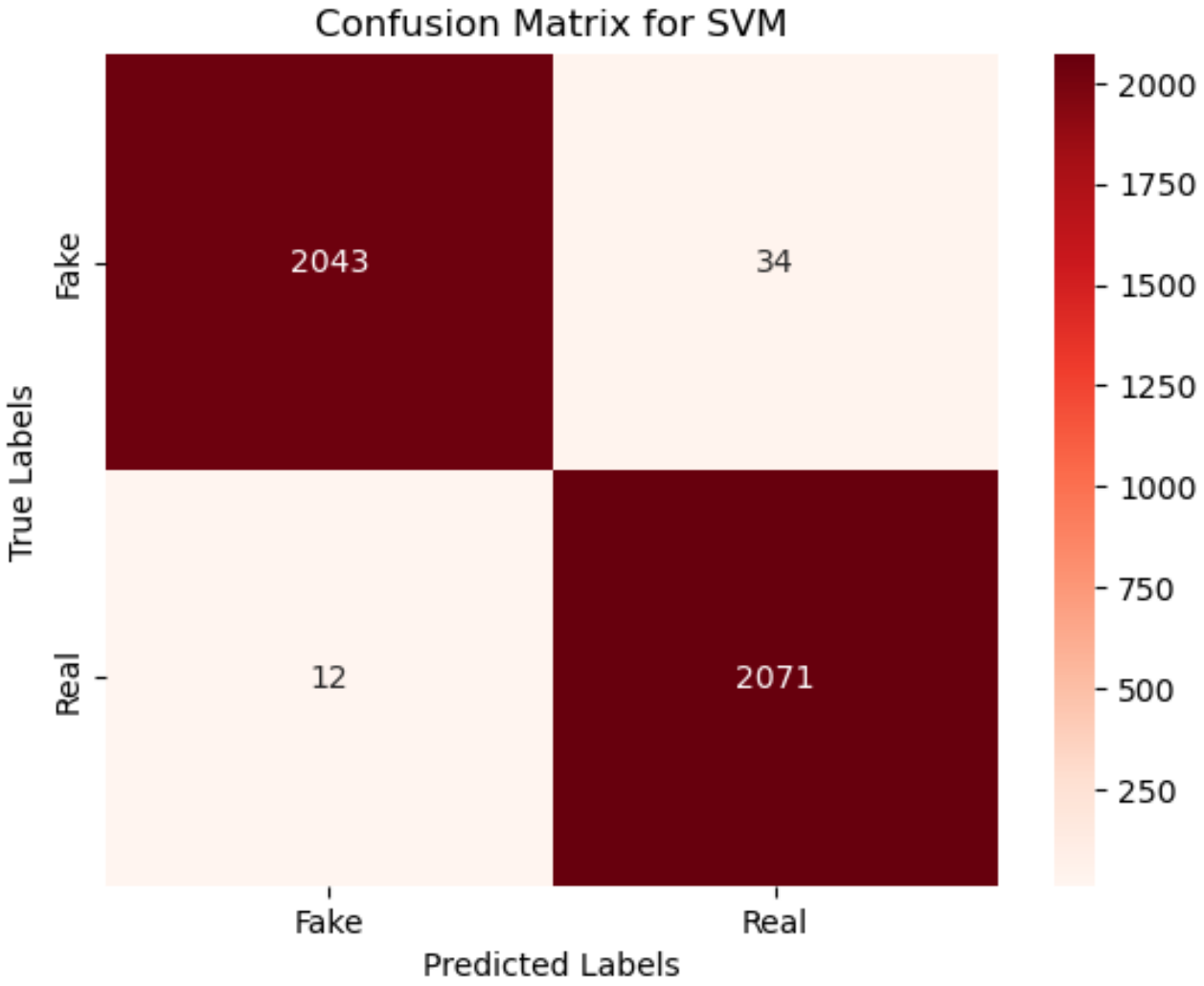
From above figure shows confusion matrix plot for logistic regression classifier.



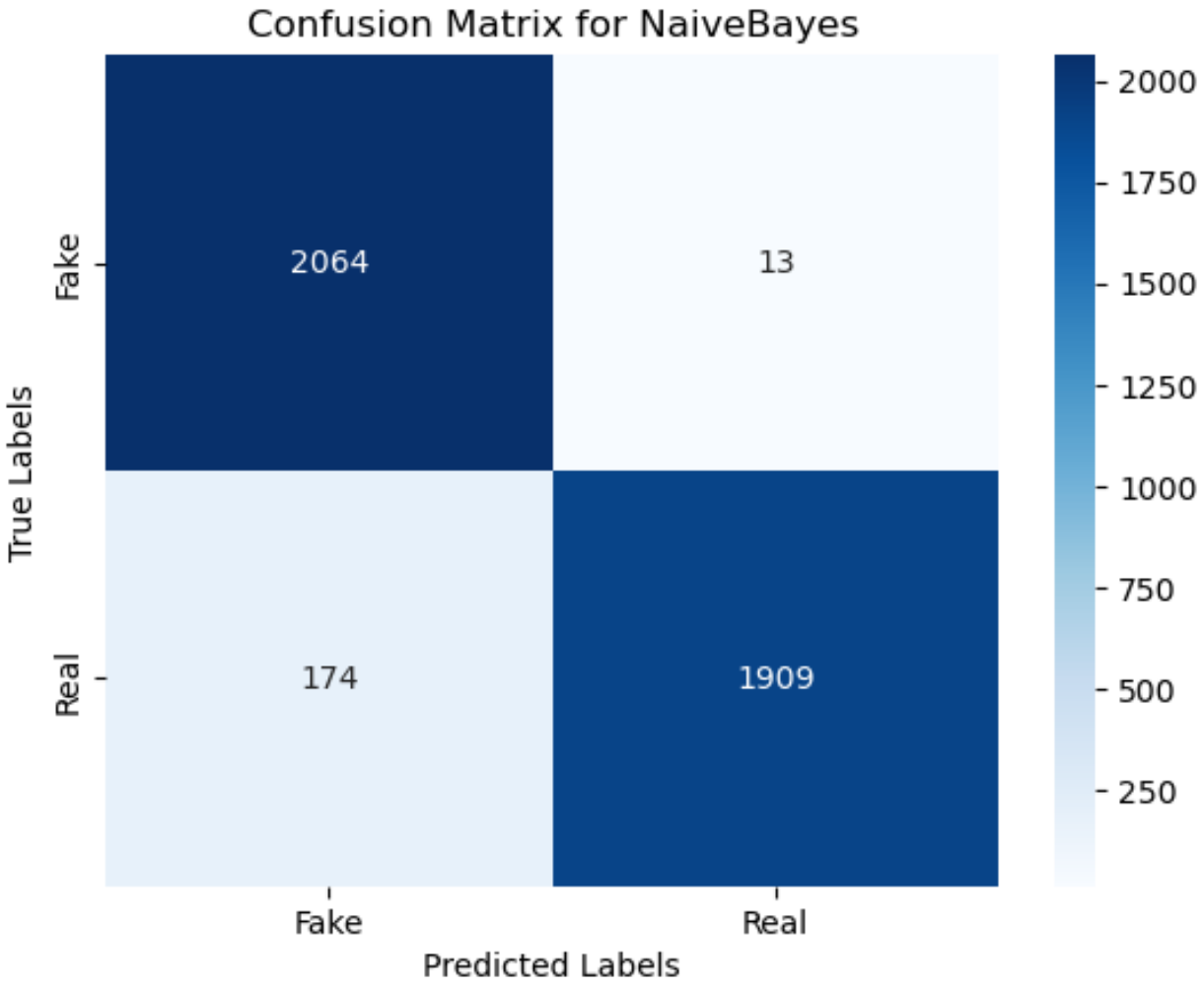
From above figure shows confusion matrix plot for Decision Tree classifier.



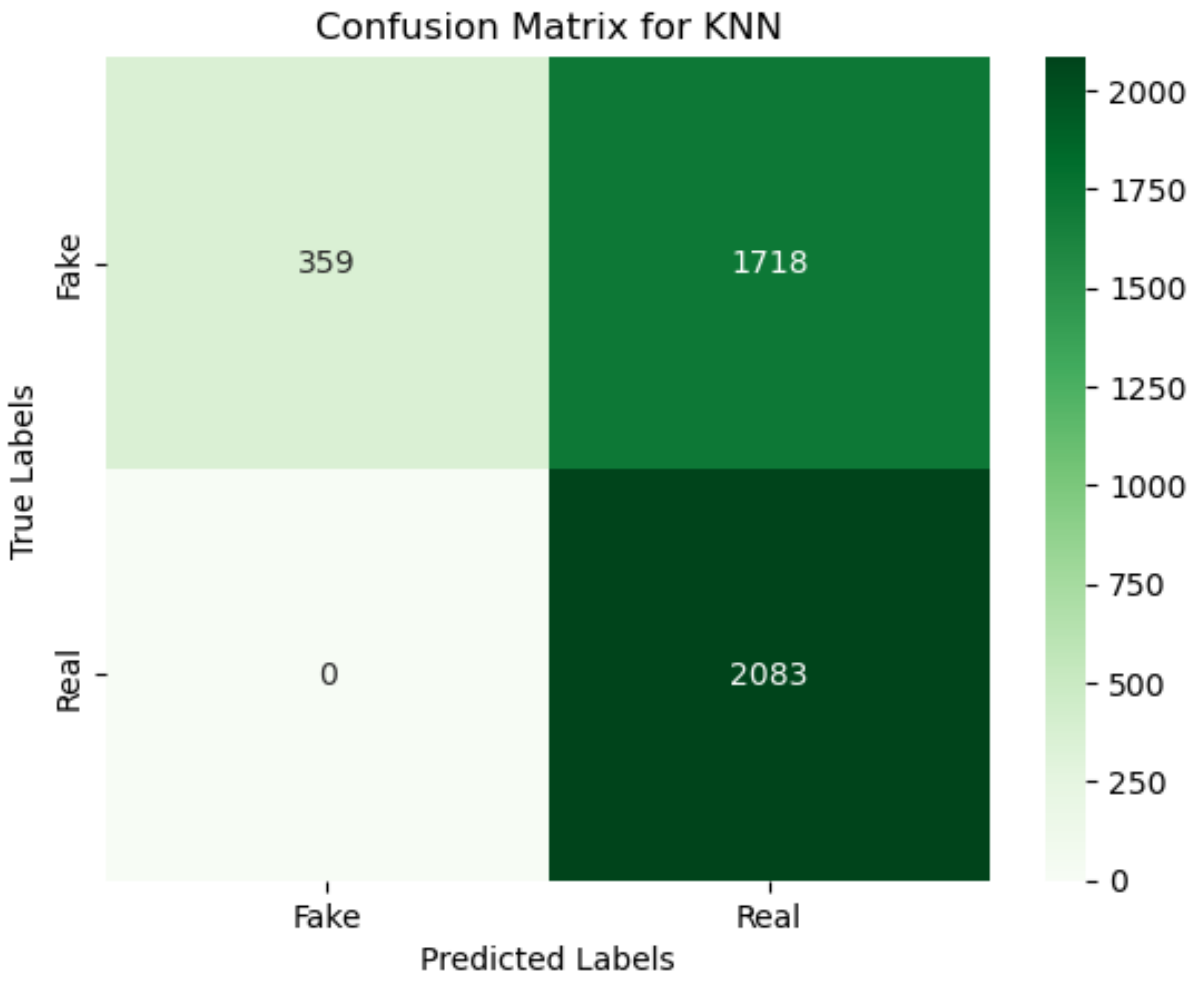
From above figure shows confusion matrix plot for Random Forest classifier.



From above figure shows confusion matrix plot for Support Vector Machine.



From above figure shows confusion matrix plot for Naïve Bayes classifier.



From above figure shows confusion matrix plot for KNN

Discussion

The results obtained demonstrate that machine learning algorithms, especially logistic regression, can be effective in detecting fake news. The high accuracy and F1 score achieved indicate that the models are capable of distinguishing between real and fake news articles with reasonable accuracy.

These results indicate that all six models perform reasonably well in detecting fake news. However, Random Forest demonstrates the highest accuracy and F1 score among the six models, making it

the most effective algorithm for this task.

However, it's important to note that the performance of the fake news detection system can be influenced by several factors. The quality and representativeness of the dataset play a crucial role in the model's performance. In this implementation, we used a dataset from Kaggle with 20,800 news articles, but it is important to consider the dataset's biases and limitations.

Text preprocessing techniques, such as removing stopwords and applying stemming, also impact the system's performance. Experimenting with different variations of these techniques or considering more advanced natural language processing approaches may further improve the model's accuracy.

Additionally, it is important to note that fake news detection is a challenging task, and the models may not capture all the nuances and complexities associated with it. False positives and false negatives may still occur, which could have real-world consequences. Therefore, the system should be continuously monitored and updated to adapt to evolving fake news patterns and improve its effectiveness.

7. CONCLUSION

We suggested a false news detection system employing machine learning methods, with an emphasis on logistic regression, in this research. We used a Kaggle dataset of 20,800 news pieces, which included text data and labels indicating whether the story was true or false. Our goal was to create a system that could successfully differentiate between legitimate and false news pieces.

We started by prepping the data by deleting unnecessary columns, dealing with missing values, and cleaning the text by removing HTML elements and special characters. To prepare the text data for analysis, we used text preprocessing techniques such as tokenization, stopword removal, and stemming.

The preprocessed text data was then used to extract important characteristics such as bag-of-words representation, n-grams, sentiment analysis scores, and readability metrics. These characteristics were used to train a variety of machine learning algorithms, including logistic regression, SVM, random forest, and decision tree.

We discovered that logistic regression attained an accuracy of 97%, precision of 96%, recall of 99%, and F1 score of 97% after training and testing the models. This suggests that the logistic regression model did a good job of recognizing bogus news.

When we compared logistic regression to various methods, we discovered that random forest had the greatest accuracy of 89%. However, logistic regression remained a viable option, offering a good combination of accuracy and interpretability.

Our project's findings illustrate the power of machine learning algorithms in spotting fake news. However, it is critical to recognize our system's limits. The system's performance can be impacted by the dataset's quality and representativeness, as well as the text preparation techniques used. False positives and false negatives are still possible, emphasizing the importance of ongoing monitoring and development.

Finally, our logistic regression-based false news detection system yields encouraging results and lays the groundwork for future study and advancement in the field. We can improve the system's accuracy and contribute to the ongoing struggle against disinformation by improving text preprocessing techniques, including more complex algorithms, and constantly updating it.

8. REFERENCES

[1] https://arxiv.org/ftp/arxiv/papers/2102/2102.04458.pdfY. Lu, Journal of Management Analytics 5, 1 (2018)

[2] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2017). Fake News Detection on Social Media: A Data Mining Perspective.

[3] Ruchansky, N., Seo, S., & Liu, Y. (2017). Leveraging Linguistic Features for Fake News Detection.
[4] Karimi, F., Dehghani, M., & Akbari, M. (2018). Combating Fake News: A Survey on Identification and Mitigation Techniques.
[5] https://iopscience.iop.org/article/10.1088/1757-899X/1099/1/012040/pdf

[6] Ma, L., Sun, G., Tang, J., & Su, Q. (2016). Detecting Rumors from Microblogs with Recurrent Neural Networks.

[7] Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., Procter, R., & Tolmie, P. (2018). Fake News Detection on Online Social Networks: A Review.

[8] Castillo, C., Mendoza, M., & Poblete, B. (2011). Detecting Fake News in Social Media Networks: A Data Mining Perspective.

[9] Raj, R. G., Pandit, A., Sharma, A., & Rani, P. (2020). Combating the Spread of Fake News: A Survey on Machine Learning Approaches.

[10] Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Fake News Detection Using Machine Learning: An Information Retrieval Perspective.

[11] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2020). Fake News Detection on Social Media: A Survey.
[12] Ruchansky, N., Seo, S., & Liu, Y. (2017). Fake News Detection: A Deep Learning Approach.
[13] Ma, L., Gao, W., Wei, Z., Lu, Q., & Wong, K. F. (2015). Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning.
[14] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2018). Combating Fake News: A Machine Learning Approach.
[15] Gupta, S., Kumaraguru, P., & Castillo, C. (2020). Fake News Detection using Deep Learning Techniques.
[16] Li, C., Xu, H., Li, S., Zhao, H., & Li, G. (2019). Fake News Detection via Multi-Source Multi-Task Learning.