FAKE NEWS DETECTION USING MACHINE LEARNING

PONNULAKSHMI

MSc STATISTICS WITH DATA SCIENCE

SREE SANKARA COLLEGE , KALADY

Period of Internship: 25th August 2025 - 19th September 2025

Report submitted to: **IDEAS** – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

In recent years, the rapid growth of online social networks has resulted in the large-scale spread of fake news for political, social, and commercial purposes. Distinguishing between fake and real information has become increasingly challenging due to the vast amount of data shared through social media platforms. Most users now consume news from social media rather than traditional outlets, increasing the risk of misinformation. Therefore, detecting fake news is a critical task to ensure the credibility of online content. In this project, we implemented Natural Language Processing (NLP) techniques for preprocessing news articles, including text cleaning and word embedding. We used Word2Vec to convert textual information into numerical representations and applied machine learning models for classification. Among the models considered, a pre-trained Random Forest Classifier was employed to classify news articles into fake or real. Data visualization methods, such as bar charts and pie charts, were also used to explore category-wise news distribution. The performance of the model was evaluated using metrics like accuracy, precision, recall, and F1-score. The results demonstrate that Word2Vec combined with Random Forest provides reliable performance in fake news detection, making it a practical solution for combating misinformation online.

2. Introduction

Information plays a vital role in shaping human decisions and influencing everyday life. In the past, news was primarily consumed through print media such as newspapers and magazines, or via electronic media like television and radio. These sources were relatively reliable as they were regulated and filtered by editorial authorities. However, with the rise of the internet and social media platforms, the public is now exposed to an overwhelming volume of unverified content. The easy access to online platforms has facilitated the spread of misinformation, fabrications, propaganda, and fake news, which can circulate rapidly across communities and influence public opinion. This has created significant challenges in society, leading to mistrust in media and difficulty in distinguishing genuine information from falsehoods.

Traditionally, the detection of fake news has relied on manual fact-checking, but this process is subjective, time-consuming, and impractical at scale. As a result, researchers and technologists have turned to Artificial Intelligence (AI), Natural Language Processing (NLP), and Machine Learning (ML) to develop automated systems for fake news detection. These technologies make it possible to process large volumes of data, analyze language patterns, and classify news articles as fake or real with greater efficiency and accuracy. Recently, the rise of Large Language Models (LLMs) such as GPT, BERT, OLMo, and Ollama has provided new opportunities in text representation, contextual understanding, and semantic analysis, making fake news detection systems even more powerful and intelligent.

The methodology followed in this project involved preprocessing the dataset using text-cleaning techniques such as lowercasing, removing special characters, and eliminating stopwords. The cleaned text was then represented using Word2Vec embeddings, which convert words into numerical vectors capturing semantic meaning. A Random Forest Classifier was used as the primary machine learning model for classification, with accuracy, precision, recall, and F1-score as evaluation metrics. Additionally, exploratory data analysis was performed using visualizations like bar charts and pie charts to understand category-wise news distribution.

The purpose of this project is to build a robust fake news detection system that can automatically classify news articles as real or fake, thereby helping to reduce misinformation and promote credible information sharing. By leveraging NLP, ML, and considering the potential of modern LLMs, the project contributes to addressing one of the most pressing challenges in the digital information age.

During the first two weeks of internship, I received training on the following topics, which provided the foundation for carrying out the project:

- Basics of Python programming for data analysis
- Data preprocessing and cleaning techniques using Pandas and Numpy
- Fundamentals of Natural Language Processing (NLP)
- Word embeddings: Bag of Words, TF-IDF, and Word2Vec
- Introduction to Machine Learning algorithms (Logistic Regression, Decision Tree, Random Forest, Naive Bayes, XGBoost)
- Large Language Models (LLMs) and tools such as Ollama for advanced NLP applications
- Model evaluation metrics (Accuracy, Precision, Recall, F1-score)
- Data visualization using Matplotlib and Seaborn
- Communication skills and professional presentation

3.Project Objective

The objectives of the project are as follows:

- To develop an automated system for detecting and classifying fake news using Natural Language Processing (NLP) and Machine Learning techniques.
- To study and analyze the linguistic and structural characteristics that differentiate fake news from real news.
- To build word embedding representations (Word2Vec) of news articles and apply a Random Forest classifier for prediction.
- To evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1-score.
- To visualize the distribution of news categories and interpret model predictions for further refinement and insights.

4. METHODOLOGY

4.1 DATASET

In this work, an open-source fake news dataset was collected from Kaggle. The dataset consists of 7 columns and 20000 rows where the columns are title, text, date, source, category, author, label.

- **title** – The headline of the news article.
- **text** – The full content of the news article.
- **category** – The category or domain of the article (e.g., politics, business, science).
- **date** – The publication date of the article.
- **source** and **author** – Metadata about the publisher and author (often missing).
- **label** – The target variable indicating whether the news is **fake (0)** or **real (1)**.

the category column contains the information about the news type, the label column contain both 0 and 1 values that represent fake and real news respectively and finally, under the text column, the text of the report has been described.

4.2 STATISTICAL SOFTWARE

In this project, Python was used as the primary statistical software for data analysis, preprocessing, model building, and evaluation. Python's extensive ecosystem of libraries enabled efficient handling of large datasets, exploratory data analysis (EDA), feature engineering, and predictive modelling. The integration of data manipulation, visualization, machine learning and statistical analysis within a single environment made Python an ideal choice for fake news detection.

The packages used are:

- **Pandas**: Used for data manipulation and analysis, providing essential data structures to clean and process the dataset. It facilitated operations such as handling missing values, transforming variables, ensuring data consistency before model training.

- **NumPy**: Utilized for numerical computations, allowing efficient handling of arrays and matrices, which are crucial for scientific computing . It was particularly useful for performing mathematical operation on datasets, computing statistical metrics, and optimizing data processing speed.

- **Matplotlib & Seaborn**: These libraries were essential for data visualization. Matplotlib provided a flexible platform for creating static visualization , while Seaborn enhanced the visual representation of statistical data. These tools were used to generate countplots, barplots,histograms, and confusion matrix, helping to uncover patterns in fake news detection.

- **Scikit-learn**: Served as the backbone for machine learning model development. It provided implementations of various algorithms such as Logistic Regression, Decision Tree, Naïve Bayes, K-Nearest Neighbors (KNN), eXtreme Gradient Boosting (XGB). Additionally, it offered tools for feature scaling, model evaluation.

- **Natural Language Toolkit (NLTK)**: Used as a comprehensive library for natural language processing tasks, including text preprocessing.

- **TextBlob**: It is a simple library that provides a simple API for text analysis tasks, including sentiment analysis, language detection and word cloud generation.

- **Imbalanced-learn (SMOTE):** Used to handle class imbalance in the dataset. Since real news cases were lower than fake news cases, SMOTE (Synthetic Minority Over-sampling Technique) was applied to generate synthetic samples for the minority class, ensuring balanced model training.(use if needed)

- **Gensim**: For training **Word2Vec embeddings** to convert words into numerical vector representations.

- **Pickle**: For loading the pre-trained Random Forest model.

  These packages collectively enabled efficient data preprocessing, in-depth exploratory data analysis, robust machine learning model development, and accurate evaluation.

4.3 DATA PREPROCESSING

To achieve better insights, it is necessary to transform the text data using preprocessing techniques, NLP, tokenization, and lemmatization before feeding them through the ML models. Data preprocessing helps to remove the noises and inconsistency of data, which increases the performance and efficiency of the model. In this work, it involves traditional techniques, regex, tokenization, stopwords, lemmatization, NLP technique, and TF-IDF/Word2Vec for data preprocessing.

The implemented data preprocessing techniques are:

- **Regex**: We use regex to remove punctuations from the text data. Often in the sentences, there may have extra punctuations like exclamatory signs. We use regex to remove those additional punctuations to make the dataset noise-free. Regex is based on context-free grammar.
- **Tokenization:** Tokenization, preprocessing tool is used to break the sentences into words .
- **Stopwords** : We use the English stopwords library in our preprocessing technique because our model data is English. We need to use the stopwords preprocessing technique to remove noises, make the model faster and more efficient, and save memory space.
- **Lemmatization**: Lemmatization is used to transform the words into root words. We can resolve data ambiguity and inflection with lemmatization. NLP techniques have been applied to convert the texts into meaningful numbers to feed these numbers into our proposed machine learning algorithm.
- **Bag of words**: The bag of words technique converts texts into machine understandable numbers, which is expressed as:

  $$TF - IDF = TF_{td}.IDF_t$$

  where t is a term, and d denotes the documents. TF stands for term frequency, which is a measurement of how frequently a term appears in a document. Consequently, term frequency TF is measured as:

  $$F = q_{td} / Number\ of\ terms\ in\ the\ document$$

  where q is the number of times the term, t appears in the document, $d.IDF$ denotes inverse document frequency, which indicates the importance of a particular term. IDF is calculated as:

$\mathrm{I}DF$ = [log(1+n)/ (1+df)dt] +1

where n means the number of documents and the denominator indicates the document frequency of the term t.

- **Feature Selection:** SelectKBest, a feature selection technique used to selects the top k features according to a specified score function. It reduces the dimensionality of high dimensional data. Here we take chi-square as score function selecting top 1000 features.
- **Synthetic Minority Over-Sampling Technique (SMOTE):** SMOTE is done to handle class imbalance problems in machine learning datasets. It is applied in target column to balance the instance. Thereby, it helps in enhancing model performances.But in this project the dataset is balanced.

4.4 EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is the process of examining, summarizing, and visualizing a dataset to understand its structure, patterns, and relationships before applying machine learning models. It helps in detecting missing values, outliers, and inconsistencies while providing insights into feature distributions and dependencies.

Key steps in EDA include:

- Descriptive Statistics: Computing measures like mean, median, standard deviation, and percentiles to summarize numerical features.
- Duplicate Removal: Identifying and removing duplicate records to ensure data integrity and prevent biased analysis,
- Data Visualization: Using histograms, count plots, pie diagrams, and word clouds to explore feature distributions and importance.
- Missing Value Analysis: Identifying and handling missing data through imputation or removal.

DESCRIPTIVE STATISTICS

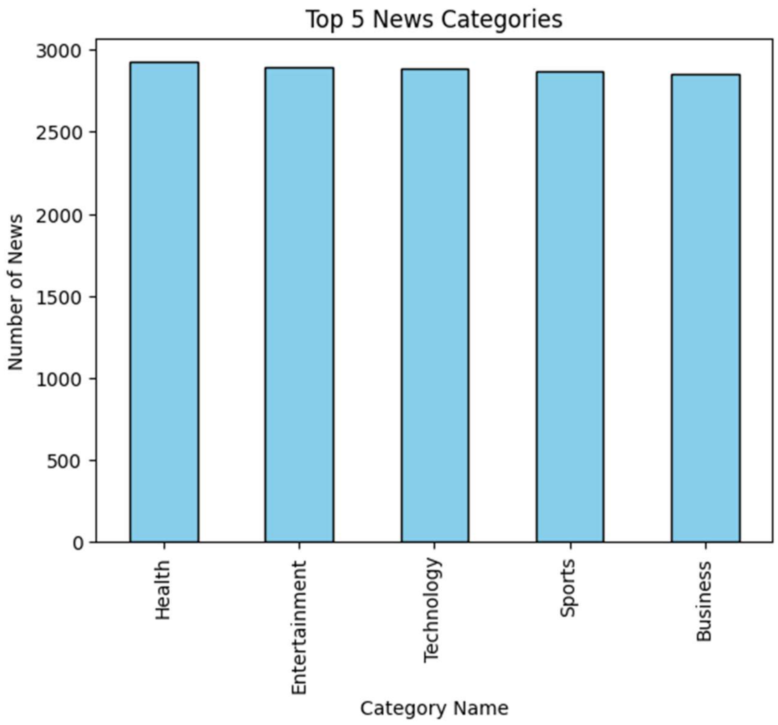
Descriptive Statistics involves summarizing and analyzing datasets using measures such as mean, median, standard deviation, and percentiles to understand data distribution and central tendencies.

DATA VISUALIZATIONS

Data visualization is the process of creating graphical representations of data to better understand and communicate insights and patterns. By leveraging various techniques, tools, and best practices, individuals can create effective data visualizations that inform decision-making, drive business strategy, and enhance understanding. As data continues to grow in volume and complexity, data visualization will play an increasingly important role in unlocking its value.

1. Bar Chart

A bar/column chart is a visualization technique used to represent categorical data with rectangular bars, where the length or height of each bar corresponds to the frequency or count of that category. It is useful for comparing values across different categories.

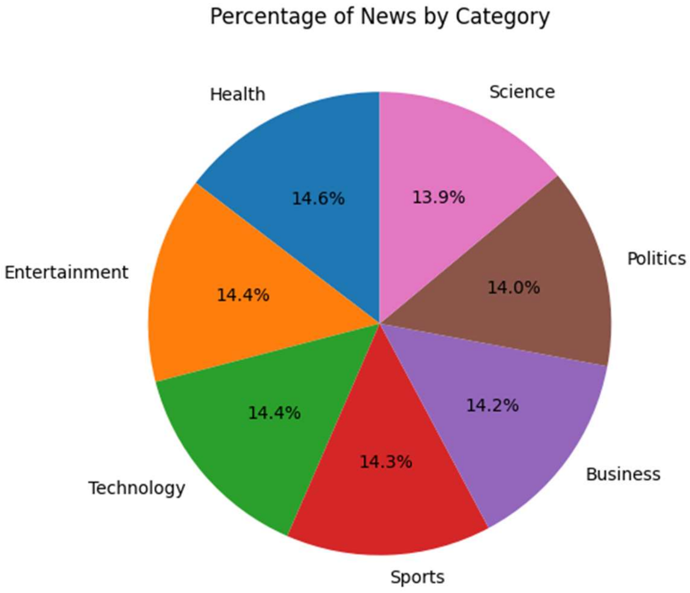


- The bar chart highlights the top 5 categories with the highest number of news articles: **Health, Entertainment, Technology, Sports, and Business**.

- Each of these categories has close to **2900 articles**, showing a uniform distribution.
- This indicates that the dataset provides strong representation across these categories, making it useful for training machine learning models to detect fake news across diverse topics.

2.Pie Chart

A pie diagram, also known as a pie chart, is a circular statistical graphic divided into slices to illustrate numerical proportion. Each slice represents a category and its size visually represents an item's proportion to the total.



- The pie chart shows the distribution of news articles across different categories.

- Categories such as **Health (14.6%)**, **Entertainment (14.4%)**, **Technology (14.4%)**, **Sports (14.3%)**, and **Business (14.2%)** are almost equally represented, with slight differences.
- **Science (13.9%)** and **Politics (14.0%)** also have nearly the same share, indicating a balanced dataset.
- Overall, the chart suggests that no single category dominates, which helps ensure that the classification model is not biased toward one particular subject.

4.5 ALGORITHMS USED FOR CLASSIFICATION

To detect and classify real and fake news, we have used different machine learning algorithms such as Random Forest Classifier, AdaBoost Classifier, eXtreme Gradient Boosting (XGB) and Logistic Regression.

1.Random Forest Classifier

Random Forest is an ensemble learning algorithm that builds multiple decision trees during training and merges their outputs to improve classification accuracy and control overfitting. Each tree is trained on a random subset of the data and features, and the final prediction is made by majority voting (for classification) or averaging (for regression).

Advantages:

- High Accuracy: Performs better than individual decision trees by reducing overfitting through ensemble averaging.
- Handles Missing Values: Can handle missing data and maintain accuracy.
- Robustness: Works well with both categorical and numerical data, and can capture non-linear patterns effectively.
- Feature Importance: Provides an estimate of feature importance, helping in feature selection.

Limitations:

- Complexity: More computationally expensive than a single decision tree due to the construction of multiple trees.
- Interpretability: Difficult to interpret compared to individual decision trees, as it is a “black-box” model.
- Large Datasets Needed: May require a large amount of training data and memory to build accurate models.

Applications:

- Fraud detection
- Sentiment analysis
- Medical diagnosis
- Fake news detection

2. AdaBoost (Adaptive Boosting)

AdaBoost is an ensemble learning method that combines multiple weak classifiers (usually decision stumps) to form a strong classifier. It works by sequentially training classifiers, where each new classifier focuses more on the misclassified instances from the previous iteration.

Advantages:

- High Accuracy: Improves performance significantly compared to a single weak classifier.
- Focuses on Errors: Assigns higher weights to misclassified samples, leading to better handling of difficult cases.
- Flexibility: Can be combined with different base classifiers, not only decision trees.
- Good Generalization: Often resistant to overfitting, especially with proper tuning.

Limitations:

- Sensitive to Noisy Data: Misclassified noisy samples may be given too much weight, reducing performance.
- Computational Cost: Requires more time to train as classifiers are built sequentially.

- Parameter Tuning: Performance depends on hyperparameters such as number of estimators and learning rate.

Applications:

- Text classification (spam detection, fake news detection)
- Face recognition
- Customer churn prediction
- Intrusion detection in cybersecurity

2. XGBoost (Extreme Gradient Boosting)

XGBoost is a highly efficient and scalable implementation of gradient boosting. It builds trees sequentially, where each new tree corrects the errors made by previous trees. It includes regularization, parallelization, and advanced optimization techniques, making it one of the most powerful machine learning algorithms.

Advantages:

- High Accuracy: Consistently outperforms many other algorithms in machine learning competitions.
- Speed and Efficiency: Optimized for parallel computation, making it faster on large datasets.
- Regularization: Includes L1 and L2 regularization to prevent overfitting.
- Flexibility: Can handle regression, classification, and ranking tasks effectively.
- Handles Missing Values: Automatically deals with missing data during training.

Limitations:

- Complexity: More complex and harder to interpret compared to simpler models.
- Computationally Expensive: Requires significant resources for very large datasets.
- Hyperparameter Tuning: Performance depends heavily on careful tuning of many parameters.

Applications:

- Fake news and text classification
- Customer churn and risk prediction
- Credit scoring and fraud detection
- Medical outcome prediction
- Recommendation systems

4. Logistic regression

Logistic regression is a widely used statistical model for binary classification problems. It is manipulated to model the probability of a certain existing event, such as true/false, reliable/unreliable, win/lose, etc. Hence, the logistic model is one of the most appropriate models for the fake news detection system. The condition for predicting logistic model is

$$0 \leq h(x) \leq 1$$

The logistic regression sigmoid function is expressed as,

$$h\theta(x) = g(\theta^T X)$$

where,

$$g(z) = 1/\ (1+x^{-z})$$

and the cost function of logistic regression is,

$$(\theta) = 1/m\sum^{m}_{i=1} (h(x^i\ ,y^i))$$

Advantages:

- Interpretability: The coefficients of the model provide insights into the relationship between input features and the output, making it easy to understand which features are most important.
- Efficiency: It is computationally inexpensive to train and predict, making it suitable for large datasets or real-time applications.

- Probabilistic Output: Logistic Regression provides probabilities for each class,which is useful for decision-making tasks where confidence levels matter.
- Works Well with Small Data: It performs well even when the dataset is small, provided the data is linearly separable.

Limitations:

- Linear Decision Boundary: Logistic Regression assumes a linear relationship between features and the log-odds of the target variable. It struggles with non-linear relationship unless feature engineering or transformations are applied.
- Sensitive to Outliers: Outliers in the dataset can significantly affect the model's performance.
- Multicollinearity: If features are highly correlated, the model may become unstable and produce unreliablr coefficients.
- Not Suitable for Complex Patterns: It cannot capture complex interactions between features without manual feature engineering.

Applications:

- Medical Diagnosis
- Customer Segmentation
- Credit Scoring

5. DATA ANALYSIS AND RESULTS

This section summarizes the outcomes of the Fake News Detection project. The analysis is divided into model evaluation, comparative analysis, and testing saved models on new datasets.

5.1 Descriptive Analysis

- The datasets contain both fake and real news articles with text, titles, categories, and labels.
- Preprocessing steps included:
    - Lowercasing
    - Removing special characters, numbers, and extra spaces
    - Stopword removal and lemmatization
    - Vectorization using Word2Vec embeddings and later TF-IDF vectorization

5.2 Evaluation Matrix

One of the crucial phases in creating a successful machine learning model is evaluating its performance. Different metrics also referred to as performance metrics or evaluation metrics are used to assess the effectiveness or quality of the model. These performance indicators enable us to evaluate how well our model handled the supplied data. By adjusting the hyperparameters, we can make the model perform better. Not all metrics can be used for all types of problems; hence, it is important to know and understand which metrics should be used. Different evaluation metrics are used for both Regression and Classification tasks. Some of the evaluation metrics for classification are:

1. Accuracy

2. Precision

3. Recall

4. F1 score

5. Confusion matrix

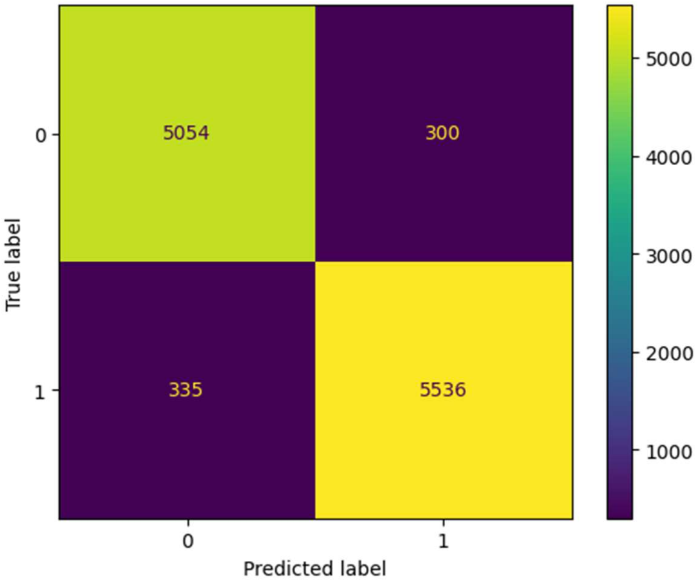
6. Classification report

5.3 Confusion Matrix

A confusion matrix is a table that summarizes the performance of a classification model by showing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. It provides a more detailed evaluation of the model's predictions and helps in assessing its accuracy and error types. For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on. The matrix is divided in to two dimensions predicted values and actual values. Predicted values are those values, which are predicted by the model and actual values are the true values for the given observations.

The confusion matrix further illustrates the model's performance by showing the number of correct and incorrect predictions for each class.

1. **Logistic Regression(Word2Vec)**



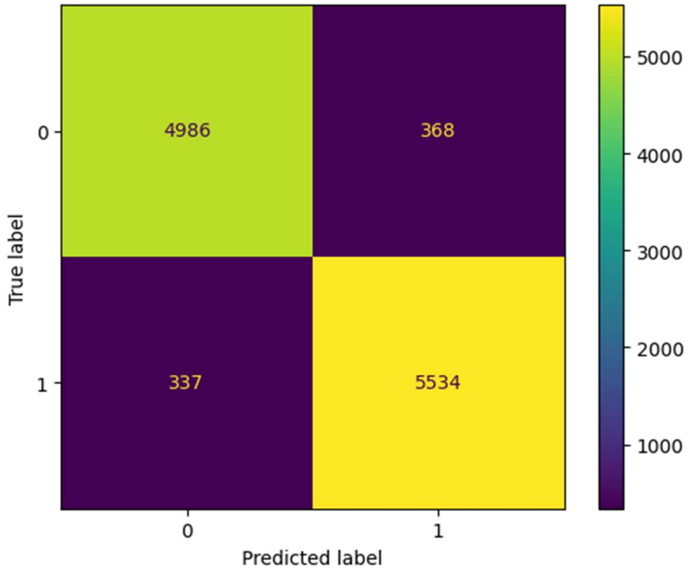
Accuracy: 0.9434298440979956

Precision: 0.9485949280328992

Recall: 0.9429398739567365

F1 score: 0.9457589476381651

2. Random Forest Classifier (Word2Vec)



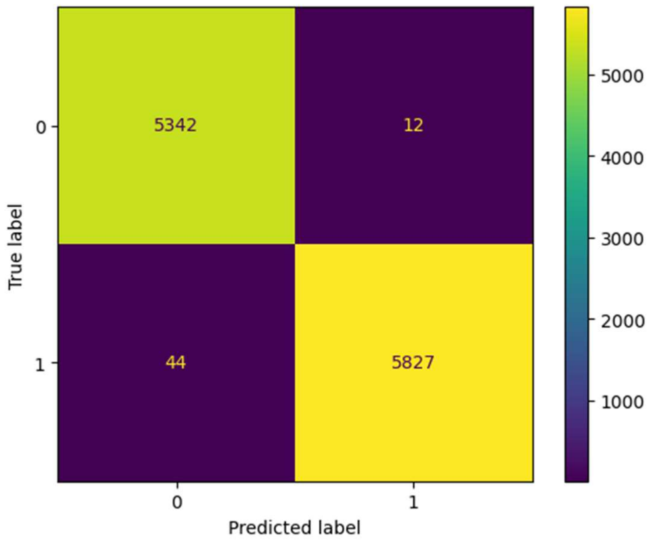
Accuracy: 0.9371937639198218

Precision: 0.9376482548288716

Recall: 0.9425992164878215

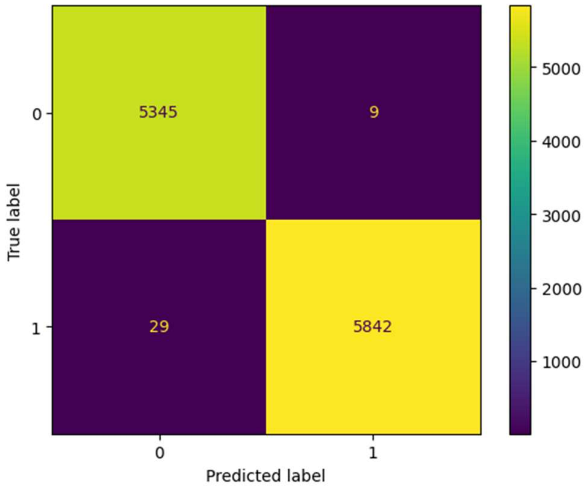
F1 score: 0.9401172173617599

3. Ada Boost (TF-IDF)



AdaBoost + TF-IDF Accuracy: 0.995011135857461

4. XGBoost (TF-IDF)



XGBoost + TF-IDF Accuracy: 0.9966146993318485

Using Saved Model on a New Dataset

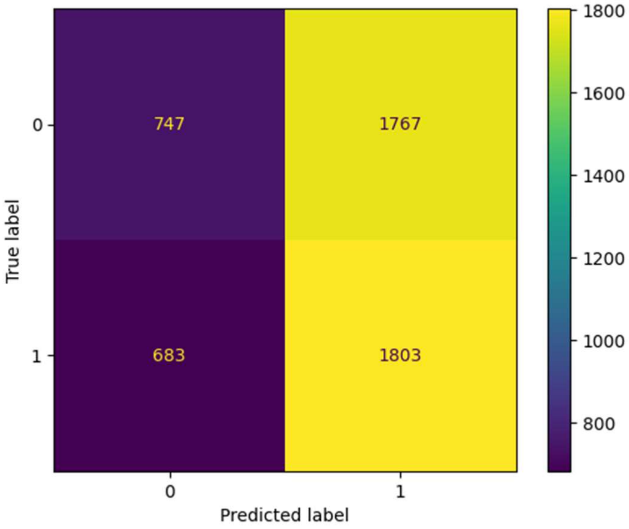
To test the generalization ability of my trained model, I saved the Random Forest classifier along with the vectorization process using Pickle. Later, I loaded this saved model into a new Colab notebook and applied it to a different dataset.

The new dataset was preprocessed in the same way as the training data (cleaning text, removing special characters/URLs, converting to lowercase). The text was then vectorized using TF-IDF, ensuring compatibility with the saved model.

Once the dataset was transformed, the Random Forest model was used to predict labels for the new news articles. Model performance was then evaluated using accuracy, precision, recall, F1-score, and a confusion matrix to understand misclassification patterns.

This experiment showed that the saved model could be reused on unseen datasets without retraining, making it efficient and practical for real-time fake news detection. However, accuracy dropped compared to the training dataset, which highlights the challenges of domain shift when applying models to new data sources.

1. Random Forest on new dataset



Accuracy: 0.51

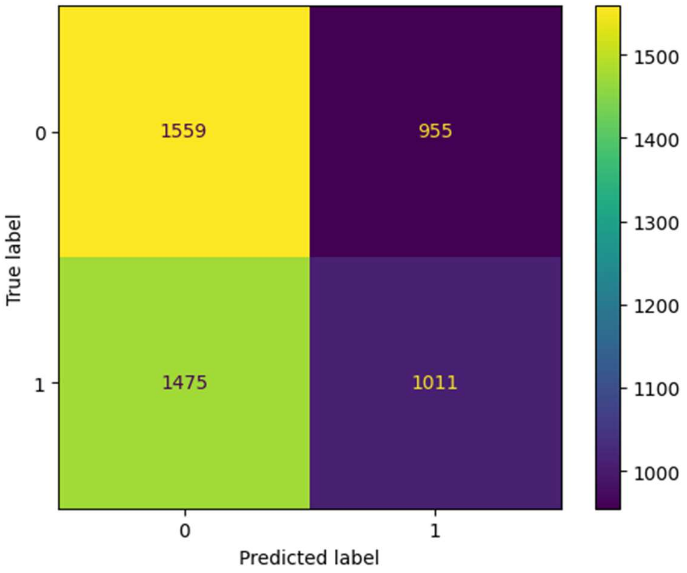
Precision: 0.5050420168067227

Recall: 0.7252614641995173

F1 score: 0.595442536327609

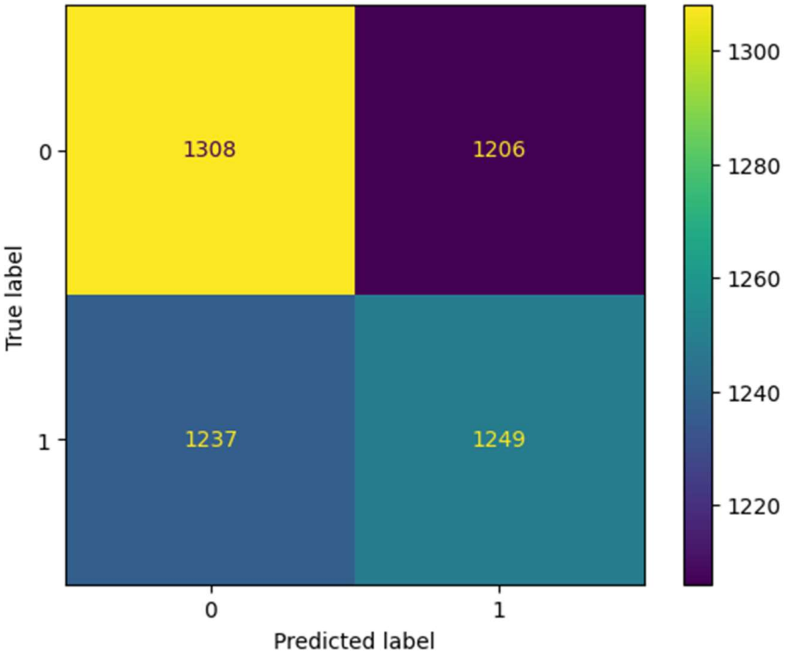
Enhancing Model Accuracy with Boosting and TF-IDF Vectorization

- **AdaBoost**



AdaBoost + TF-IDF Accuracy: 0.514

- **XGBoost**



XGBoost + TF-IDF Accuracy: 0.5114

6. Conclusion

The project on Fake News Detection has provided important insights into the role of machine learning in combating misinformation. By systematically preprocessing the dataset, applying text vectorization techniques like TF-IDF and Word2Vec, and experimenting with multiple supervised learning models, the study was able to build, compare, and analyze models for distinguishing fake news from real news.

Among the models trained on the primary dataset, Logistic Regression achieved an accuracy of 94.34% with balanced precision, recall, and F1-score, making it a strong and interpretable baseline. Random Forest also performed competitively, reaching an accuracy of 93.72%. However, the use of boosting techniques with TF-IDF vectorization significantly enhanced model performance. AdaBoost with TF-IDF achieved 99.50% accuracy, and XGBoost with TF-IDF further improved it to 99.66%, confirming the superiority of ensemble-based methods when combined with appropriate text representation.

When evaluating generalization by applying the saved Random Forest model to a new dataset, accuracy dropped to 51%, with relatively high recall but lower precision. This indicates that while the model could still detect many fake articles, its performance suffered due to domain differences between the training and new dataset. Similar trends were observed when boosting models with TF-IDF were applied to the new dataset, where AdaBoost and XGBoost achieved around 51% accuracy, highlighting the challenge of model adaptability in real-world scenarios.

From these results, it can be concluded that ensemble models, particularly boosting methods with TF-IDF vectorization, are the most effective for fake news detection on the training dataset, achieving near-perfect classification. However, the performance drop on unseen datasets demonstrates that models trained on curated datasets may struggle to generalize to different sources of news. This underlines the importance of building robust, adaptable systems that can handle diverse and evolving misinformation.

Looking forward, improvements can be made by training on larger and more diverse datasets, incorporating multilingual and real-time data sources, and exploring advanced deep learning models such as LSTMs, GRUs, or transformer-based architectures like BERT. Additionally, integrating contextual features such as author credibility, publishing source, and temporal patterns could enhance detection reliability.

In summary, this project validates the potential of machine learning, especially ensemble techniques, in detecting fake news with high accuracy. At the same time, it emphasizes the need for stronger generalization strategies and adaptability to ensure that such systems remain practical and effective in real-world applications.

7. APPENDICES

7.1 References

1. Kaur, Sawinder & Kumar, Parteek & Kumaraguru, Ponnurangam. (2020). Automating fake news detection system using multi-level voting model. Soft Computing. 24. 10.1007/s00500-019-04436-y.
2. Ko, J., Lee, S., & Kim, H. (2019). Fake news detection system using reverse tracking approach.Information Processing & Management.
3. de Oliveira, N.R.; Medeiros, D.S.V.; Mattos, D.M.F. A Sensitive Stylistic Approach to Identify Fake News on Social Networking. *IEEE Signal Process. Lett.* **2020**, *27*, 1250–1254.
4. Park, M.; Chai, S. Constructing a User-Centered Fake News Detection Model by Using Classification Algorithms in Machine Learning Techniques. *IEEE Access* **2023**, *11*, 71517–71527.
5. J. Stromback, Y. Tsfati, H. Boomgaarden, et al., “News media trust and its impact on media use: Toward a framework for future research,” Annals of the International Communication Association, vol. 44, pp. 139-156, 2020.
6. Parikh, S. B., & Atrey, P. K. (2018, April). Media-Rich Fake News Detection: A Survey. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 436-441). IEEE.
7. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015, November). Automatic deception detection: Methods for finding fake news. In Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community (p. 82). American Society for Information Science.
8. Fatmeh Torabi Asr, Maite Taboada:“Big Data and quality data for Fake News and misinformation detection”, Big Data & Society, 2019, Article-14, DOI:10.1177/2053951719843310.
9. PranavBharti,MohakBakshi,R.AnnieUthra:“FakeNewsDetectionUsingLogistic
10. Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.