Team : Ones and Zeroes
Machine Learning

MACHINE LEARNING

Fake News detection using `Python` and Machine Learning

Shawn Sam Varghese [1], Jobin Tom[2], Kevin Thomas[3] and Siju K S[4,]

[1]Saintgits College of Engineering (Autonomous) and [1]Institutional Mentor, Intel-Unnati Programme

[*]shawns.csb2125@saintgits.org ; jobint.csb2125@saintgits.org; kevinaddis101@gmail.com; siju.swamy@saintgits.org
[†]All are members of the team Ones and Zeroes

Abstract

Online social networks have seen a surge in fake news, leading to widespread dissemination for commercial and political gain. Fake news manipulates information cleverly, making it contagious among social media users and impacting offline society. To create a trustworthy online environment, this project focuses on timely identification of fake news articles, creators, and subjects. It also evaluates the performance of detection mechanisms. The proposed approach involves a machine learning model to predict whether news articles are real or fake, utilizing various techniques to enhance accuracy. By leveraging machine learning, we can effectively combat misinformation and promote an informed digital society.

Introduction

We are entering a new world. The technologies of machine learning, speech recognition, and natural language understanding are reaching a nexus of capability. The end result is that we'll soon have artificially intelligent assistants to help us in every aspect of our lives. -Amy Stapleton

Fake news online is a pressing challenge, undermining the reliability of information. Deceptive narratives flood social media, eroding trust and manipulating opinions. Researchers leverage machine learning and data analytics to develop robust detection methods, combating misinformation and restoring trust. Challenges include evolving tactics, diverse datasets, and ethical considerations. Progress in this field offers hope for a resilient information landscape.

This project explores approaches, methodologies, and advancements in fake news detection. By understanding the complexities of this issue, we aim to create a more trustworthy information ecosystem. Our research focuses on leveraging machine learning and data analytics to identify and address fake news effectively. Despite challenges such as evolving tactics and ethical dilemmas, we are dedicated to mitigating the impact of misinformation and safeguarding the integrity of our society.

Methodology

i. **Data collection** This stage refers to the process of gathering and obtaining relevant data to train a machine learning model. We can gather data required for our project from various web scraping, articles, headlines, metadata, and associated contents. We efficiently use these data sets to train and test our model using it's diverse data.

ii. **Data cleaning and pre-processing** In this step we remove irrelevant or duplicate data, such as advertisements, boilerplate content, or non-textual elements. Standardize the text by converting to lowercase, removing punctuation, and handling special characters. Handle misspellings, abbreviations, and stemming to reduce noise in the text. Remove stop words (commonly occurring words with little semantic value). Perform text normalization, such as dates, and numerical values.

Compiled on: July 12, 2023.
Draft manuscript prepared by the author.

iii. **Data encoding** After data cleaning and pre-processing has been completed we encode categorical features, such as source, author, or category, using one-hot encoding or label encoding. The other non-textual data are numerical representations suitable for ML models.

iv. **Vectorization** The pre-processed text data is converted into numerical representations for ML models to process. Various techniques such as bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), word embeddings (e.g., GloVe or Word2Vec), document embeddings, or character-level encoding can be applied to accomplish this. In this project TfidfVectorizer from sklearn has been used.

v. **Training, Evaluation and Improving machine learning models** We Split the dataset into training and testing sets for model evaluation and selecting appropriate ML algorithms, such as logistic regression, naive Bayes classification model, decision tree and passive aggressive classifier. The ML models are trained based on the labeled training dataset. The models are then evaluated using performance metrics such as accuracy, precision, recall, and F1 score. Based on the evaluation we further fine-tune the models by adjusting hyperparameters or using techniques like cross-validation.

. After the above procedures have been concluded the trained model is ready to be deployed and classify new or unseen news articles as real or fake.

Implementation

In the project, we use Python script to implement fake news detection using machine learning algorithms.The code begins by importing the necessary libraries, including *pandas, seaborn, matplotlib, nltk, os,* and *pickle.* We use Intel optimised library scikit learn to improve the performance of all the models. These libraries are essential for data manipulation, visualization, natural language processing, and model serialization.

After which the code proceeds to download the data set from URL https://shorturl.at/jmnU7 and https://www.kaggle.com/datasets/sonalgarg174/ifnd-dataset. The downloaded files are then unziped in the directory *data*. The unzipped file contains the datasets *Fake.csv* and *True.csv*. The dataset collected gives 2,1519 data entries for *True* news and 23,492 data entries for *Fake* news, totally giving 45,011 entries.

After obtaining the datasets we use the pandas library to seperate these files into separate datasets containing columns emphtitle, text, subject, date and *label* , where the label coloumn is used to determine whether the news is fake(0) or true(1).

For getting a better understanding of the data we visualise it using seaborn and matplotlib. It creates count plots to display the distribution of fake and real news, as well as the distribution of news across different subjects.

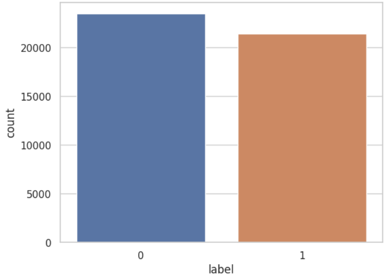

Figure 1. Visualisation of inputed Dataset

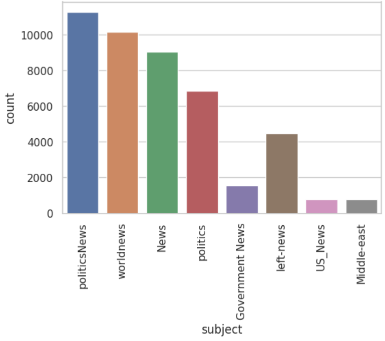

Figure 2. Caption for Figure 2

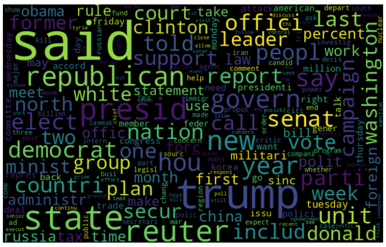

Figure 3. Caption for Figure 3

Moving on to data preprocessing, we remove the unrequired columns - title, subject and date. We also concatenate the title and text columns into a single column named text. The code also checks for any null values in the dataset using a custom function called *Check_forNAN*, after which the data set is shuffled randomly. The data pre-processing is completed within 365.119 seconds for our prepared dataset.

Before moving on the text data requires cleaning. To accomplish this, we use several functions. The *tokenize* function tokenizes the text into individual words using the nltk library. The *remove_stopwords* function eliminates common English stopwords from the tokenized words. The *apply_stemming* function applies stemming to the remaining words using the PorterStemmer algorithm. Finally, the *rejoin_words* function rejoins the stemmed words into a single string. These functions are then applied to the *text* column of the dataframe, resulting in a new column named *rejoined*.

Next, we use the *TfidfVectorizer* from sklearn to convert the text data into a numerical representation. The data is then split into training and testing sets using the *train_test_split* function from sklearn. This allows for model training on the training data and evaluation on the testing data.

The machine learning models Logistic Regression, Naive Bayes, Decision Tree, and Passive-Aggressive Classifier models are employed for training, evaluation, and prediction. For each model, we test, compare and calculate accuracy scores using confusion matrices and ROC curves to assess their performance. The Decision Tree model is selected due to it's better performance and serialized using pickle, saving it to a file named "selected_model.pkl."

Finally we define a function called *fake_news_det* that uses the saved model to predict the label of new news articles and classify news articles as fake or real. The code and the readme file for this project are available at the GitHub repository : https://github.com/jobint001/Fake_news_detection Results of this implementations is discussed in the below section section.

Result and Discussion

The project has been successful in achieving the goal of creating a fake news detection facility to overcome the proliferating fake news in this digital age. We have trained various Machine Learning models and evaluated them to select the best preferred model. The Machine Learning models usen in this project are Logistic Regression, Naive Bayes classification model, Decision Tree, and Passive-Aggressive Classifier.

We have used confusion matrices and ROC graphs to help us further evaluate the models. The confusion matrix provides detailed information about the model's performance, giving insights into the true positive, true negative, false positive, and false negative predictions of the models. while the ROC curve illustrates the trade-off between the true positive rate and the false positive rate at different classification thresholds. The confusion matrices and ROC graphs obtained from evaluating the models are given below,

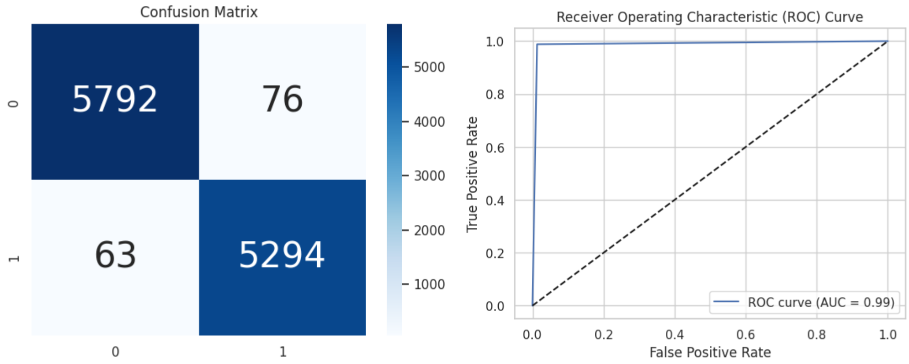

Figure 4. Logistic Regression

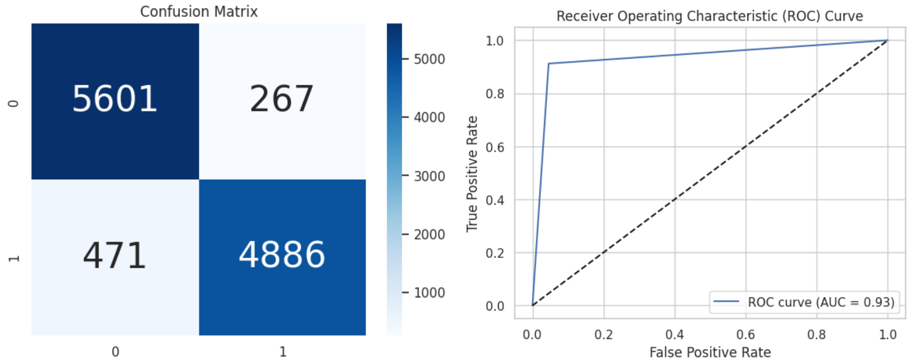

Figure 5. Naive Bayes classification model

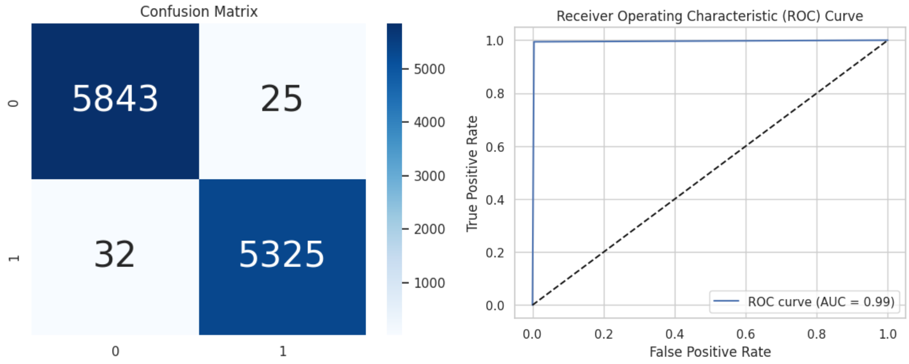

Figure 6. Decsision Tree

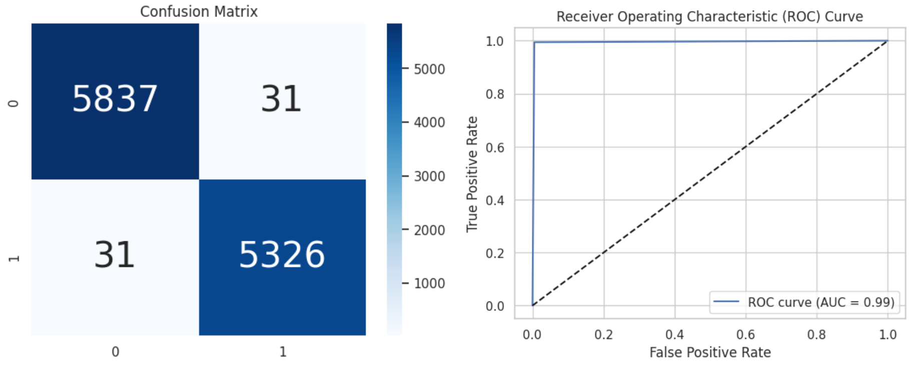

Figure 7. Passive-Aggressive Classifier

The summary of the of the various evaluations and results of the confused matrices and ROC curves of the Machine Learning models have been compiled and shown in Table 1.

The four different machine learning models were implemented and evaluated as follows,

- The Logistic Regression model achieved an accuracy score of 98.80%, showcasing its effectiveness in predicting the authenticity of news articles. It utilized a linear regression algorithm to classify the data.
- The Naive Bayes classification model achieved an accuracy score of 93.79%, indicating its ability to classify news articles as real or fake. It showed promising performance in distinguishing between the two categories.
- The Decision Tree model achieved an accuracy score of 99.58%, providing another viable approach to fake news detection. It demonstrated its capability to create decision rules based on features derived from the dataset. It demonstrated good performance in classifying news articles.
- Lastly, the Passive-Aggressive Classifier model achieved an accuracy score of 99.51%, highlighting its ability to adapt to new data and make online updates to its model.

Table 1. Performance of Machine Learning Models in testing the fake news data

Model No.	Model Name	RMSE value	Accuracy	Precision	f1-score	Recall	Processing Time
1	Logistic Regression	0.110	98.80	0.988	0.988	0.099	2.694 sec
2	Naive Bayes	0.249	93.79	0.938	0.938	0.937	0.35 sec
3	Decision Tree	0.065	99.58	0.996	0.996	0.996	21.878 sec
4	Passive Aggressive	0.070	99.51	0.995	0.995	0.995	29.890 sec

From the above results we can conclude that the Decision Tree model demonstrated a high true positive rate and true negative rate, indicating its ability to correctly classify both real and fake news articles. This model shows promise for identifying and detecting fake news accurately. Future development could focus on incorporating additional features and exploring more advanced machine learning techniques to further enhance the accuracy and reliability of fake news detection models.

Conclusions

In this project, we investigated the effectiveness of machine learning models (Logistic Regression, Naive Bayes, Decision Tree, and Passive-Aggressive Classifier) for fake news detection. The Decision Tree model demonstrated the highest accuracy, precision, and F1-scores among the tested models, while the Passive-Aggressive Classifier showed comparable performance but with slightly slower processing time. These findings contribute to the field of fake news detection and highlight the potential of these models in combating misinformation and promoting reliable information. However, further research is needed to explore ensemble methods, advanced feature engineering techniques, and domain-specific knowledge integration to address the challenges associated with detecting nuanced forms of fake news.

In conclusion, our study demonstrates the effectiveness of Logistic Regression, Naive Bayes, Decision Tree, and Passive-Aggressive Classifier models for fake news detection. The Decision Tree model, in particular, shows promise with its high accuracy and precision. Enhancements through advanced techniques and continued research are necessary to improve the models' performance and tackle the complexities of identifying various forms of fake news. Developing robust and accurate fake news detection systems is crucial for preserving information integrity and enabling informed decision-making in the digital era.

Acknowledgments

We express our heartfelt gratitude to Intel© for the invaluable industrial training opportunity provided through the Intel© - Unnati Programme at Saintgits College of Engineering. We sincerely appreciate the guidance and support from the Intel© team throughout the training. Our Project Guide and Institutional Mentor, Mr. Siju K S, deserves special thanks for his valuable guidance, suggestions, and constant encouragement, which led to the successful completion of our project and enhanced our industry understanding. We also thank the management and faculty of Saintgits College of Engineering for facilitating this opportunity. Being a part of the Intel© - Unnati Programme has been an honor, and we are confident that the skills acquired during this training will shape our future career. [0]

References

1. Vijayarani S, Janani R, et al. Text mining: open source tokenization tools-an analysis. Advanced Computational Intelligence: An International Journal (ACII) 2016;3(1):37–47.
2. Singh J, Gupta V. Text stemming: Approaches, applications, and challenges. ACM Computing Surveys (CSUR) 2016;49(3):1–46.
3. Wilbur WJ, Sirotkin K. The automatic identification of stop words. Journal of information science 1992;18(1):45–55.
4. Sial AH, Rashdi SYS, Khan AH. Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python. International Journal 2021;10(1).
5. Teoh TT, Rong Z. Text Mining. In: Artificial Intelligence with Python Springer; 2022.p. 227–237.
6. Zou X, Hu Y, Tian Z, Shen K. Logistic regression model optimization and case analysis. In: 2019 IEEE 7th international conference on computer science and network technology (ICCSNT) IEEE; 2019. p. 135–139.
7. Singh M, Bhatt MW, Bedi HS, Mishra U. Performance of bernoulli's naive bayes classifier in the detection of fake news. Materials Today: Proceedings 2020;.
8. Lyu S, Lo DCT. Fake news detection by decision tree. In: 2020 SoutheastCon IEEE; 2020. p. 1–2.
9. Gupta S, Meel P. Fake news detection using passive-aggressive classifier. In: Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020 Springer; 2021. p. 155–164.

Project Code In Implementing Fake News Detection

Github repo

https://github.com/jobint001/Fake_news_detection