BAA1027

Data Analytics: Advanced Python and Machine Learning

Whitney Airewele

21368531

Word count: ~3,366

Introduction

Background on Covid-19 Misinformation

Since the beginning of the Covid-19 pandemic, it has been accompanied by an infodemic which Ferreira Caceres et al., (2022) discusses as the age of fake news and misinformation which has spread resulting in undermining public health efforts to share factual news, scientific guidance, and vaccination efforts. The reason these work and people give into fake news is because "they provide the comfort of an explanation in times of uncertainty and anxiety" (The Lancet Diseases, 2020). Additionally, the author mentions that because messaging revolves around core emotions and values it hijacks the mental cues that are used to decide whether the source is legitimate and thus trustworthy. The false information spread around the topic of Covid-19 has ranged from unfounded treatment recommendations to conspiracy theories surrounding its origin. Additionally, it has perpetuated many different and dangerous beliefs such as mask refusal, unproven treatments and vaccine refusal which has led to increased morbidity (Ferreira Caceres., 2022; Roozenbeek et al., 2020). Overall, it is clear, fake news and misinformation weakens trust and reliance on institutions trying to tackle the problem.

Problem Statement

With the rapid increase in the spread of fake news, computational techniques such as fact checking websites such as PolitiFact and Snopes have been developed as ways to mark certain articles as fake or real based on their textual content (Ahmad, 2020). However, the author also outlines limitations in this, noting that "the problem with these resources is that human expertise is required to identify articles/websites as fake." Additionally, these websites are not generalised (Ahmad, 2020). This has highlighted the critical need for automated deception systems capable of identifying potentially false news. While the human fact checking techniques are valuable, the sheer amount of data out there just demands for more effective approaches to be used. Machine learning techniques offer a solution for this problem particularly when applied to text based news classification. This is because "a number of studies have primarily focused on detection and classification of fake news on social media platforms such as Facebook and Twitter" (Ahmad, 2020).

Project Objectives and Dataset

The dataset as briefly mentioned includes labeled data collected from various news sources thus eligible for supervised learning. Initial data exploration revealed a significant class imbalance where fake news (0) was in the majority class. The imbalance ratio was 20.5:1 presenting an additional methodological challenge that required specific handling techniques like oversampling.

- This project aims to develop and evaluate a machine learning approach to classify Covid-19 news headlines as either fake (class 0) or real (class 1).
- Implement text processing suitable for machine learning models to interpret.
- Compare and determine the best model performance.
- Address the dataset imbalance problem.
- Identify the textual features strongly associated with indicating whether news is fake or real.
- The research questions guiding this are which classification algorithm performs best? Which oversampling technique was most effective and how does this impact model performance on imbalanced data?

This report provides a methodology section, results analysis, and discussion. Topics mentioned include data preprocessing, feature engineering, model selection and evaluation metrics. The report compares the performance of Decision Tree, Random Forest, and Logistic Regression models before and after implementing oversampling. The discussion critically evaluates these results as well as reflects limitations in the code with potential improvements that could also help further research.

Literature Review and Related Works

As mentioned, the rapid spread of false information surrounding Covid-19 has led to massive concern in discerning what is real or fake. With human expertise having its limitations, researchers have found more effective and efficient approaches to do this through machine learning algorithms like those assessed in this project.

Recent studies have explored feature extraction and text processing and its importance in fake news detection. Felber (2021) explored this by experimenting with “various steps like stop word removal, stemming/lemmatisation, link removal and more”. By applying classical machine learning algorithms with diverse linguistic features including n-grams, readability metrics, emotional tone, and punctuation the author was able to achieve a 95.19% F1 score using a linear SVM model. Similarly in the article by Baarir and Djeffal (2020) they proposed a system for fake news detection in which they made use of Term Frequency- Inverse Document Frequency (TF-IDF) of bag of words for information retrieval and natural language text processing. The authors also utilised n-grams as a feature extraction technique which was found to be more effective than the bag of words technique. The classifier was SVM which they concluded as the best algorithm to detect fake news.

The code implementation in this project aligns with the work of Baarir and Djeffal (2020) by similarly making use of TF-IDF. Although they found this to be less efficient compared to n-grams, the effectiveness of its use has been consistently demonstrated across multiple studies. Khanam et al. (2021) used this method for their implementation and their best performing models (XGBoost at 75% accuracy, SVM and Random Forest at ~73%) were achieved using this feature extraction method as opposed to other count-based approaches.

Literature suggests a range of different algorithms for fake news detection. The common approaches used for this sort of classification are SVM or tree-based models. These approaches seem to yield the most accurate results as seen in Felber (2021) achieving a 95.19% F1 score while Al-Ahmad et al. (2021) reported SVM achieving 73% accuracy. However, Felber (2021) next best performing model was logistic regression thus supporting its implementation in this project. Likewise, according to Kumar and Arora (2021) logistic regression has been employed successfully in several fake news detection studies often generating strong comparison. In a study they referred to, logistic regression performed well on Kaggle sourced datasets. As well as this, despite random forest being Felber (2021) least performing model it still produced quite high accuracy and F1 score. Its use in studies has demonstrated stable performance. Coca et al. (2018) found random forest had the best overall accuracy (95.93%) followed by SVM. While Al-Ahmad et al. (2021) proposed an evolutionary approach using feature selection with random forest, they achieved a 75.4% accuracy. Their evolutionary approach suggests avenues for future

exploration and optimisation. This project also includes decision tree implementation which Thair Ali et al. (2024) found to outperform random forest in terms of the classification accuracy, but random forest was better suited to large datasets. This may be because of decision trees being prone to overfitting. Although hyperparameter tuning does help in balancing model complexity. Hence grid search is used. All the literature cited further highlights the importance of dataset selection when evaluating classification models. The code implementation includes handling a class imbalance with oversampling. Lastly, the code also evaluates models using metrics such as accuracy, precision, recall, F1-score, and roc auc thus keeping consistent with the literature. Overall, it suggests that the approach used for this study should produce competitive outcomes especially with random forest.

Methodology

The dataset sourced from Kaggle in this study consists of news headlines which are the input features labeled as real (1) or fake (0), the outputs. It focuses on developing a machine learning solution to detect Covid-19 fake news in online media. With this becoming an increasingly prevalent issue the objective was clear. To use a classification model to distinguish this based on textual content. The dataset itself contains 10,202 rows with each entry noted under a binary classification label as mentioned above. This section will essentially explain the step-by-step implementation of the code for this fake news classifier, focusing on why specific methods were chosen and how they contributed to the final model (random forest with SMOTE).

First initial exploration was done to obtain information on the dataset and see it loaded in spyder. Then significant class imbalance was identified (20.5:1) and then visualised in a graph. It revealed that for every real news article there were approximately 20.5 fake news articles. Due to classification algorithms usually favouring the majority class, this posed an issue and justified the need for oversampling to be used. Without such, models I trained were bound to be biased and display high overall accuracy while underperforming in the minority class. This would cause the model to miss real news (false positives) therefore not leading to the best outcomes especially in fake news detection.

<u>Data Preprocessing</u>

The steps involved in this section consisted of transforming the data from the csv file into an appropriate manner for the machine learning model to digest. The first action once necessary libraries were downloaded was to clean the text. Here I addressed missing values and duplicates. Additionally, during the project the news headlines required natural language processing (NLP) techniques. This was used for text normalisation so essentially converting all text to lowercase to ensure consistency and prevent identical words being treated differently (e.g., “Covid” and “covid”). As well as this punctuation, special characters, numbers, and excess whitespaces were stripped to help in standardising the text. To further process the data tokenisation was implemented to split the text into individual words to help analysis. These NLP techniques were critical because news headlines often include patterns that can distinguish whether it is authentic or fake. By doing this I was able to collect these indications better.

For feature extraction, TF-IDF vectorisation converted text to numerical featured using the following:

```
#feature extraction using TF-IDF and also converts text to numerical for transformation
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X = tfidf_vectorizer.fit_transform(df['cleaned_headlines'])
y = df['outcome']
```

This was chosen because it gives less importance to common words that do not help distinguish whether an article is real or fake. Several studies support this such as Suhasini and Vimala (2021) who highlight that TF-IDF is an efficient feature extraction method that performs better than bag of words models. Similarly, Kumar (2020) emphasises its efficiency with machine learning models used for fake news detection. This achieved up to 94% accuracy and outperformed CountVectoriser. Additionally, I considered whether structural characteristics of headlines (text length) might correlate with or play a role in determining fake from real news. Feature engineering was integrated into this through text length analysis.

Regarding data splitting, the first split created an 80/20 split (80% for training and validation, 20% for testing). While the second splits the 80% portion with a test size of 0.25, resulting in:

- 80% × 0.75 = 60% for training
- 80% × 0.25 = 20% for validation

- 20% for testing

Stratification was used to prevent subsets with skewed distributions and ensure the same class distribution.

As mentioned, there was a significant class imbalance among the data. To address these two oversampling techniques were applied:

- Random oversampling which randomly duplicated samples from the minority class.
- SMOTE which is used to create synthetic samples of the minority class.

These techniques were just applied to the training set, so the validation and test sets were able to reflect real world distribution like in the imbalanced data. Through this it was able to show which improves performance on this data.

Model selection and Implementation

The three classification algorithms used to compare performance were:

- Decision Tree
- Random Forest
- Logistic Regression

Firstly, the decision tree was chosen due to its simplicity. It has previously been assessed in these sorts of studies and has not performed well thus I saw this as room for further research. Next, random forest an ensemble method was applied as it is less prone to overfitting and has improved accuracy. Lastly, logistic regression which similar to random forests has had quite high performances across studies provided a linear approach with the outputs probability based. This enabled a fair evaluation of diverse models for this project.

Model Training and Optimisation

The selected models were trained under three main conditions:

1. The original imbalanced data
2. The data with random oversampling
3. The data with SMOTE

Then for hyperparameter tuning grid search with three-fold cross validation was carried out. The parameters included:

- For Decision Trees: maximum depth, minimum samples split, minimum samples leaf, and criterion.
- For Random Forest: number of estimators, maximum depth, minimum samples split, and minimum samples leaf.
- For Logistic Regression: regularization strength (C), penalty type, and solver.

In terms of model optimisation, the F1 metric was used because the data was imbalanced thus accuracy was misleading. With that, the F1 scores provided a mean between precision and recall which was necessary as both false positives and false negatives have notable consequences.

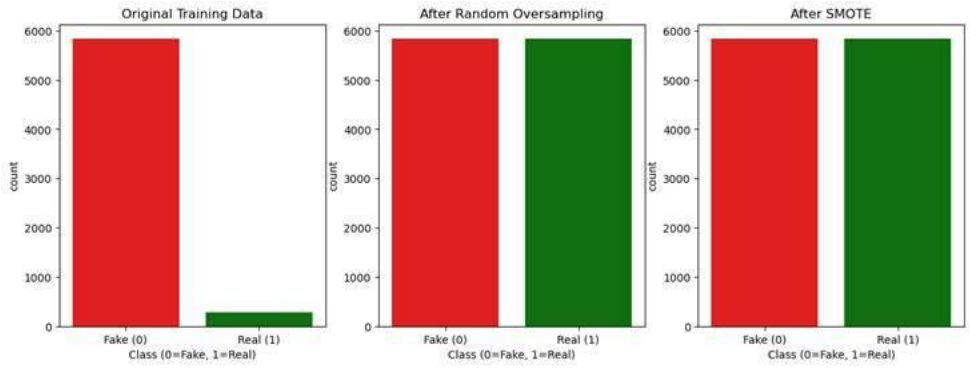
Evaluation Metrics

Overall to evaluate model performance for these classification algorithms, I used accuracy, precision, and recall. F1 score and roc-auc. Performance was then all visualised through confusion matrices for each sample set and roc curves. For further study, feature importance analysis was applied to identify the most influential words.

Results

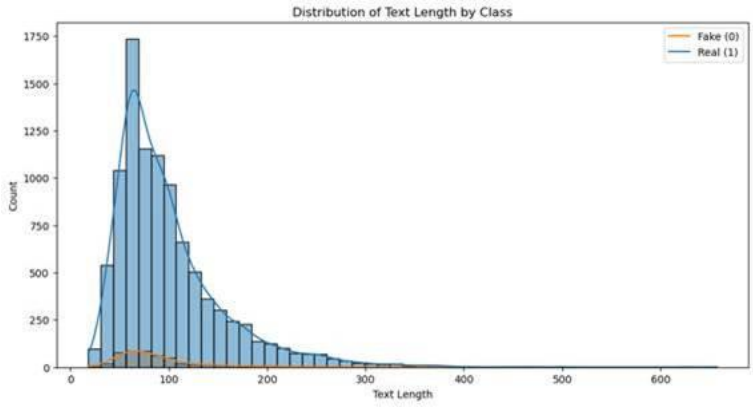
Addressing class imbalance

Looking at initial exploration, this revealed quite a large class imbalance of 20.5:1. Therefore applying oversampling techniques was necessary. Random oversampling and SMOTE able to bring the minority class up to a balance of a 1:1 ratio:



Text analysis

The analysis of text length revealed that while it is not a definite predictor of legitimacy, textual structure could be important for classification. Real news headlines displayed a clear bell-shaped distribution peaking at around 80-90 characters. In contrast fake news was much flatter and dispersed. The dramatic difference suggested that real news sources were more consistent in headline length while fake news had a slight bit more variability.



Model performance comparison

Performance was evaluated on the original imbalance data, random oversampling and SMOTE:

1. Original data

Model Performance on Original Imbalanced Data

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Decision Tree	0.9559	0.5253	0.5474	0.5361	0.7616
Random Forest	0.9735	0.9767	0.4421	0.6087	0.8717
Logistic Regression	0.9618	0.9474	0.1895	0.3158	0.9070

All models achieved high accuracy however this is misleading due to the imbalance. Thus, recall for the minority class struggled. The random forest demonstrated the best overall performance with their F1 score.

2. Random oversampling

Model Performance with Random Oversampling (ROS)

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Decision Tree	0.9426	0.4154	0.5684	0.4800	0.7647
Random Forest	0.9755	0.9592	0.4947	0.6528	0.8722
Logistic Regression	0.9657	0.6316	0.6316	0.6316	0.9117

Once random oversampling was applied model performance changed. Once again random forest maintained the F1 score and roc auc. Logistic regression also showed improvement.

3. SMOTE

Model Performance with SMOTE

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Decision Tree	0.9471	0.4435	0.5368	0.4857	0.7520
Random Forest	0.9770	0.9615	0.5263	0.6803	0.8979
Logistic Regression	0.9657	0.6344	0.6211	0.6277	0.9083

SMOTE processing achieved the highest performance at a 0.6803 F1 score and 0.9770 accuracy with random forest. Logistic regression had a balanced performance between precision, recall and F1 score.

4. F1 score comparison

F1 Scores Comparison Across Sampling Techniques

Model	Original	Random Oversampling	SMOTE
Decision Tree	0.5361	0.4800	0.4857
Random Forest	0.6087	0.6528	**0.6803**
Logistic Regression	0.3158	0.6316	0.6277

Key findings:

- Random Forest with SMOTE achieved the highest F1 score (0.6803)
- Both oversampling techniques improved Random Forest performance
- Logistic Regression showed significant improvement with oversampling
- Decision Tree performance actually decreased with oversampling

Hyperparameter tuning

Once random forest with SMOTE was identified as the best model, hyperparameter tuning was utilised to optimise its performance. This was done using GridSearchCV with a three fold cross validation for each of 108 different parameter combinations totaling 324 fits. The below image shows the optimal configuration which allowed trees to grow to their maximum depth of thirty, leaf notes to contain as few as one sample, split them with as few as two samples and use maximum two hundred trees in the ensemble. The best cross validation F1 score was 0.9970,

suggesting that model performance changes across different data splits which are common with imbalanced datasets.

```
elif best_val_model == 'Random Forest':
    print("\nTuning Random Forest Hyperparameters")
    param_grid = {
        'n_estimators': [50, 100, 200],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]
    }
    base_model = RandomForestClassifier(random_state=42)
```

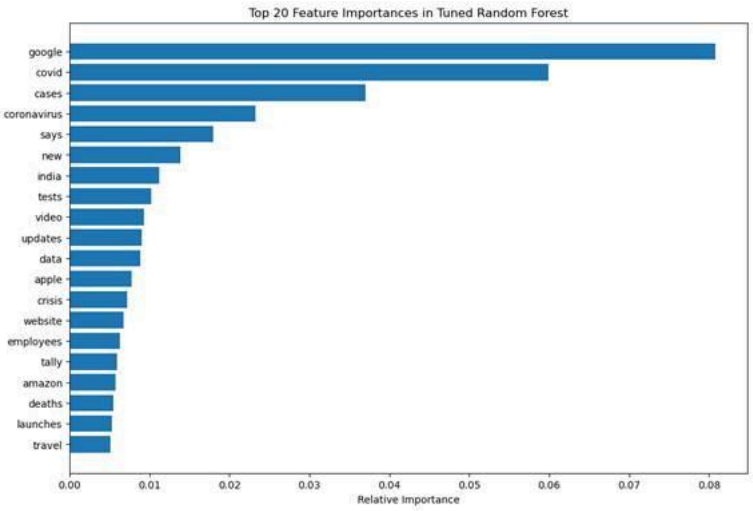
Tuned Random Forest Results:

- Accuracy: 0.9770
- Precision: 0.9615
- Recall: 0.5263
- F1 Score: 0.6803
- Roc Auc: 0.8979

These results were tuned from random forest SMOTE suggesting this is its optimum.

Feature importance

An analysis of this revealed the top twenty words which influenced model decisions.



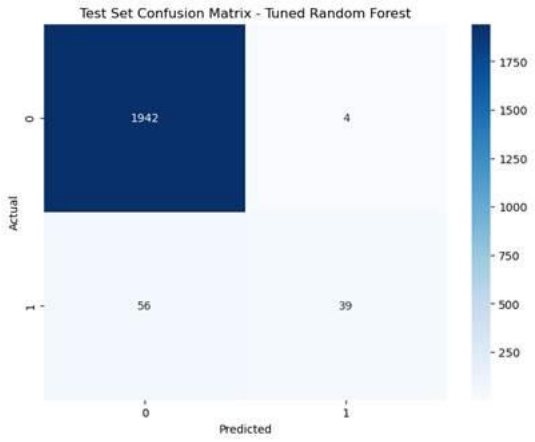
Final evaluation

Once tested, the best model (random forest SMOTE) is the following:

Final Test Results for Best Models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Random Forest (SMOTE)	0.9706	0.9070	0.4105	0.5652	0.9043
Tuned Random Forest	0.9706	0.9070	0.4105	0.5652	0.9043

As we can see, performance across all metrics did not improve on the test set for both tuned and SMOTE models except for roc auc. The confusion matrix gives more insight into this model's classification performance tuned and untuned:



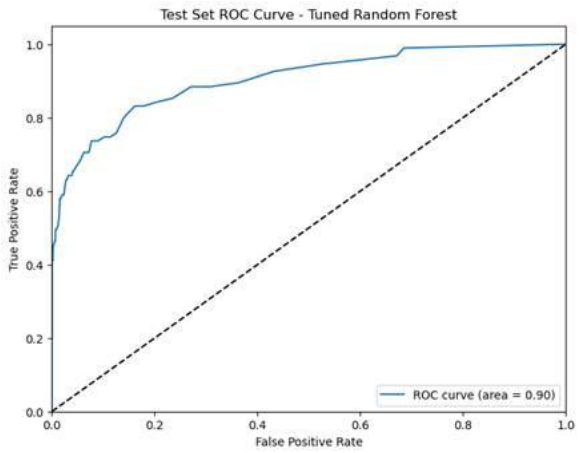
Confusion Matrix for Best Model (Random Forest with SMOTE)

	Predicted Fake (0)	Predicted Real (1)
Actual Fake (0)	1942	4
Actual Real (1)	56	39

True positives (correctly identified real news): 39 articles
False negatives (real news classified as fake): 56 articles
True negatives (correctly identified fake news): 1942 articles
False positives (fake news classified as real): 4 articles

Despite using SMOTE during training, the model still shows bias to the majority class of fake news. The low number of fake news classified as real (false positives) means that when the model predicts an article is real is more dependable (precision 0.9070). While the higher number of false negatives at 56 demonstrates that it misses a lot more of the real news articles (recall 0.4105).

The roc curve further illustrates the relationship between the true positive and false positive rate. With a roc auc of approximately 0.90, it shows good ability to distinguish between real and fake. However, logistic regression performed slightly better with 0.91.



This model is best when it is important that any news classified as real is actually real. But it is still prone to missing some of these articles thus in a case where this is a significant issue the model would need to be improved.

Discussion

Consistent with literature, random forest outperformed decision trees and logistic regression. This is because random forest combines multiple decision trees to reduce overfitting and variance. The difference in F1 score results also helped in understanding why ensemble methods are often preferred for text classification. This observation is supported by Kamateri and Salampasis (2025), who further noted that "ensemble learning can improve predictive performance compared to the performance of any of its constituents alone" in their study on ensemble methods for classification. Moreover, their research showed that ensemble systems consistently achieved significant improvements over individual classifiers especially when dealing with imbalanced datasets similar to my Covid-19 fake news one. This reinforced the initial idea I had once I did some research.

The methodology followed a structured format dealing with text processing first which was crucial for ensuring consistent feature extraction which using TF-IDF was implemented to capture the importance of words relative to frequency. Additionally, studies showed that it was more effective as opposed to a simple bag of words. As vastly mentioned throughout the report, oversampling techniques were applied to address the class imbalance. Both overall improved the random forests performance but what was later surprising was that despite oversampling it still struggled with recall. The decision to evaluate decision trees, random forest, and logistic regression similar to my ideas for this project came from relevant literature. I wanted to see if decision trees had the ability to perform better while testing to see out of random forest or logistic regression which would be superior as they both had consistent results across studies. Hyperparameter tuning was then conducted to optimise performance as the right combination can improve accuracy, reduce overfitting, speed up training and boost overall performance on unseen data. This was found to be maximum depth: None, minimum leaf samples: 1, minimum sample split: 2 and n estimators: 100. As mentioned, both oversampling improved F1 scores of random forest with SMOTE proving slightly better:

- Random Forest (Original): 0.6087
- Random Forest (ROS): 0.6528

- Random Forest (SMOTE): 0.6803

The tuned model was the same for SMOTE but once each were tested and validated, they decreased in performance to 0.5652. This suggested that synthetic samples may not have improved generalisation or possibly introduced noise.

The confusion matrix highlighted that the model still struggled. This suggested that even with these techniques handling class imbalance remains a challenge. In addition to F1 scores, precision, recall and roc auc were evaluated. Although accuracy was more than 94% across all evaluations, it was misleading. Therefore, the other metrics helped provide a more comprehensive understanding. Precision for when false positives are a problem, recall for detecting minority class and roc auc for further performance assessment.

Feature importance analysis showed which words have the most influence on the models' decisions. Words such as “covid” were expected due to the focus of the dataset being Covid-19 words such as “google” returning as the highest indicator for feature importance was surprising. Possibly, the mentioning of technology companies like such suggests they are used differently in fake vs real news headlines. This proposes an area for further research.

Conclusion

This study overall achieved its objectives by identifying random forest with SMOTE as the best model for detecting Covid-19 fake news. Although the precision and moderate to low recall still indicate that minority class detection remains difficult. Lastly, as mentioned in addition to feature importance revealing an avenue for future research, the project as a whole could suggest further investigating more advanced ensemble methods and feature engineering to improve performance. Deep learning for these classification tasks is something that could be explored as it has made numerous appearances in studies such as by Alghamdi, Lin and Luo (2022) who noted its strengths.

Bibliography

Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. (2020). Fake News Detection Using Machine Learning Ensemble Methods. *Complexity*, *2020*(1), 8885861. https://doi.org/10.1155/2020/8885861

Al-Ahmad, B., Al-Zoubi, A. M., Abu Khurma, R., & Aljarah, I. (2021). An Evolutionary Fake News Detection Method for COVID-19 Pandemic Information. *Symmetry*, *13*(6), Article 6. https://doi.org/10.3390/sym13061091

Alghamdi, J., Lin, Y., & Luo, S. (2022). A Comparative Study of Machine Learning and Deep Learning Techniques for Fake News Detection. *Information*, *13*(12), Article 12. https://doi.org/10.3390/info13120576

Baarir, N. F., & Djeffal, A. (2021). Fake News detection Using Machine Learning. *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-Being (IHSH)*, 125–130. https://doi.org/10.1109/IHSH51661.2021.9378748

Coca, L.-G. (2018). *IDENTIFYING FAKE NEWS ON TWITTER USING NAÏVE BAYES, SVM AND RANDOM FOREST DISTRIBUTED ALGORITHMS*. ConsILR2018_v220191114-18260-185ie9v-libre.pdf

Felber, T. (2021). *Constraint 2021: Machine Learning Models for COVID-19 Fake News Detection Shared Task* (arXiv:2101.03717). arXiv. https://doi.org/10.48550/arXiv.2101.03717

Ferreira Caceres, M. M., Sosa, J. P., Lawrence, J. A., Sestacovschi, C., Tidd-Johnson, A., Rasool, M. H. U., Gadamidi, V. K., Ozair, S., Pandav, K., Cuevas-Lou, C., Parrish, M., Rodriguez, I., &

Fernandez, J. P. (2022). The impact of misinformation on the COVID-19 pandemic. *AIMS Public Health*, *9*(2), 262–277. https://doi.org/10.3934/publichealth.2022018

Kamateri, E., & Salampasis, M. (2025). An Ensemble Framework for Text Classification. *Information*, *16*(2), Article 2. https://doi.org/10.3390/info16020085

Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021). Fake News Detection Using Machine Learning Approaches. *IOP Conference Series: Materials Science and Engineering*, *1099*(1), 012040. https://doi.org/10.1088/1757-899X/1099/1/012040

Kumar, R. (2020). *Fake News Detection using Passive Aggressive and TF-IDF Vectorizer*. *07*(12). IRJET_V7I12158-libre.pdf

Kumar, S., & Arora, B. (2021). A Review of Fake News Detection Using Machine Learning Techniques. *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 1–8. https://doi.org/10.1109/ICESC51422.2021.9532796

Roozenbeek, J., Schneider, C. R., Dryhurst, S., Kerr, J., Freeman, A. L. J., Recchia, G., van der Bles, A. M., & van der Linden, S. (2020). Susceptibility to misinformation about COVID-19 around the world. *Royal Society Open Science*, *7*(10), 201199. https://doi.org/10.1098/rsos.201199

Thair Ali, N., Falih Hassan, K., Najim Abdullah, M., & Salam Al-Hchimy, Z. (2024). The Application of Random Forest to the Classification of Fake News. *BIO Web of Conferences*, *97*, 00049. https://doi.org/10.1051/bioconf/20249700049

The Lancet Infectious Diseases. (2020). The COVID-19 infodemic. *The Lancet. Infectious Diseases*, *20*(8), 875. https://doi.org/10.1016/S1473-3099(20)30565-X

Appendices

Appendix A: Code written in spyder

ML code spyder

Appendix B: Dataset

data.xlsx - Google Sheets