COMP3222 Coursework 2023/24

Apishan Thayananthan (at16g21)

January 12, 2024

1 Introduction and Data Analysis

1.1 Problem Description

The growing prevalence of misinformation in the digital era, particularly on social media platforms, has necessitated the development of robust algorithms for distinguishing the authenticity of shared content—the implications in an age where information can spread rapidly. False news can negatively impact public opinion, political landscapes, and personal beliefs. Hence, the task is to develop machine learning models that accurately classify social media posts as ‘real’ or ‘fake’. The ability to automatically verify the integrity of online content is invaluable for journalists, researchers, and the general public.

The definition of ‘fake’ news is as follows. Posts of Real Multimedia, where real media are misleadingly repurposed to represent current occurrences. Such misuse can lead to misinformation as the original context of the media is changed. Digitally Manipulated Content: This category includes posts that are edited. The manipulation might be subtle or extensive, but it misinterprets the truth or reality depicted in the original media. Synthetic Media involves completely fabricated content, such as artworks or computer-generated images, falsely presented as genuine real-world imagery.

This project is mainly focussed on the textual components of social media posts rather than the images themselves. The approach entails applying machine learning techniques to the text of tweets to enhance the detection of posts spreading fake news. The analysis and model building will mainly utilise Natural Language Processing (NLP) techniques, specifically forms of vectorisation. By analysing the text, the algorithm aims to distinguish patterns or indicators that can reliably classify posts as real or fake.

1.2 Dataset Characterization

1.2.1 Format

The dataset comprises social media posts, primarily including text content. Each entry in the dataset typically includes fields like tweetText, userId, username, timestamp, and label (indicating if the post is ’real’ or ’fake’). Additional metadata such as tweetId and imageId(s) is also present.

1.2.2 Volume

The dataset comprises a training dataset and a test dataset. The training dataset comprises 14277 posts, and the test dataset comprises 3755 posts. So, in total, the dataset comprises 18032 posts. Each post is labelled ’real’, ’fake’ or ’humour’. For this algorithm, the posts with the ’humour’ label were treated as ’fake’ posts.

1.2.3 Quality

The quality of the textual data in our dataset was diverse and replicated the challenges of real-life social media content. This is because the textual data contained URLs; whilst they can be informative, they were unnecessary. Especially as the task was to analyse the text, the URL information was unattainable; hence, it was excessive noise. The data also consisted of special characters, ranging from emojis to punctuation marks, introducing additional complexity. While they enrich the data with emotional and contextual cues, they challenge standard text processing techniques.

Furthermore, the dataset displays a wide range of languages, as there are 44 different languages in it. This diversity reflects the authentic nature of social media communication but also complicates the process of text normalisation and feature extraction. The meaning can be lost through translation, so the text was not translated.

The accuracy of the labels assigned to the dataset is equally critical to our analysis. Each entry is categorised as either 'real' or 'fake', and the reliability of these labels is paramount to the integrity of our model training and subsequent evaluations.

1.2.4 Bias

The dataset shows a notable inequality between the number of 'fake' and 'real' labels, with a higher prevalence of 'fake' posts. The number of posts labelled 'fake' was 9356, whereas only 4921 were labelled 'real'.This imbalance is a critical factor as it can introduce bias.

Machine learning models tend to favour the majority class—in this case, 'fake' posts. There will be more misclassification of 'real' posts, as the model is more exposed to and thus better at identifying characteristics of 'fake' posts. Understanding and addressing class imbalance is crucial for developing a model that accurately represents and predicts both 'fake' and 'real' posts. The goal is to ensure the model is accurate, fair, and effective in identifying content across different classes.

In summary, having more 'fake' labels than 'real' ones in the dataset introduces specific biases and challenges. The analysis and model development process is tailored to acknowledge and counteract these issues, leading to more reliable and equitable outcomes.

1.2.5 Detailed Analysis of Real and Fake Posts in Training Data

Sentiment Analysis

- **Real Posts:** The average sentiment score of real posts is 0.0987, indicating a slightly positive tone overall. This could be reflective of the nature of genuine content in the context of social media.

- **Fake Posts:** In contrast, fake posts have a lower average sentiment score of 0.0653, suggesting a more neutral or less positive tone. This difference in sentiment could potentially be a distinguishing feature between real and fake content.

Review Length Analysis

- **Real Posts:** The average length of real posts is approximately 51.75 words. This metric provides a baseline for the typical length of genuine social media posts in our dataset.

- **Fake Posts:** Fake posts are slightly shorter on average, with about 48.00 words. The difference in length, though subtle, might be indicative of the nature of the content.

Readability Scores

- **Flesch Reading Ease:** Real posts have a slightly higher ease of reading score (56.67) compared to fake posts (56.28), which might suggest differences in the complexity or clarity of the text.
- **Gunning Fog Index:** The Gunning Fog scores, which assess the complexity of the text, are 8.87 for real posts and 9.64 for fake posts, indicating fake posts tend to be slightly more complex or harder to read.

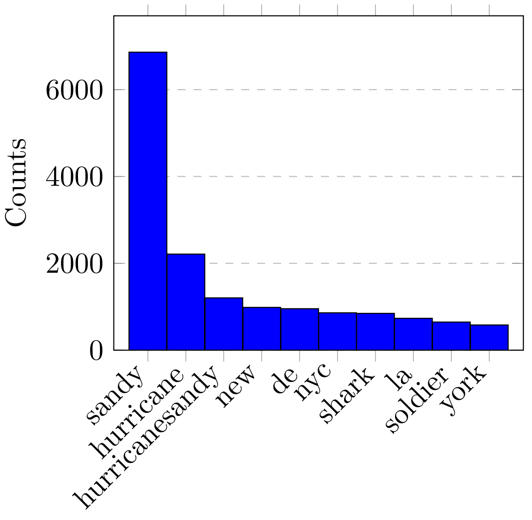

Figure 1
Top 10 most common words in Fake News Tweets

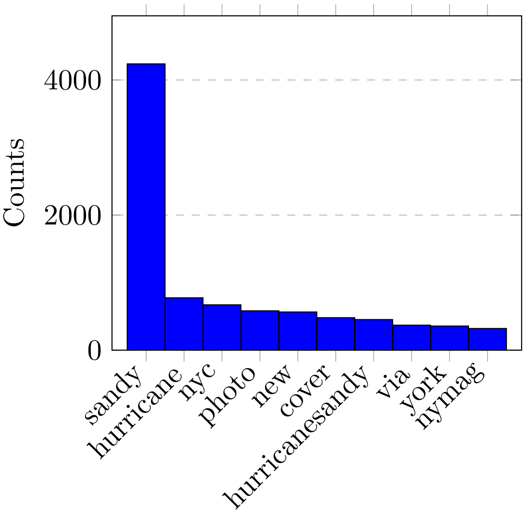

Figure 2
Top 10 most common words in Real News Tweets

Frequent N-grams Analysis: In the analysis of the most frequent words in ‘fake’ and ‘real’ news tweets, the unigrams in real posts—such as “sandy”, “hurricane”, and “nyc”—are indicative of genuine discussions centred on Hurricane Sandy and its effects on New York City. On the other hand, in fake posts, words like “sandy” and “hurricane” appear alongside attention-grabbing terms like “shark”, indicating that while these posts may reference a real event, they often do so by adding sensational or untrue elements to the narrative. The comparison reveals that while both real and fake posts may discuss the same event, the language used in fake posts tends to be more provocative, possibly aiming to exploit the viral nature of dramatic content. Machine learning models can leverage this distinction in word usage to enhance the accuracy of classifying social media content.

2 Pipeline Design

The pipeline is engineered to transform raw social media posts into a format suitable for machine learning classification. This transformation journey is crucial, ensuring the data fed into the algorithms is clean, relevant, and structured.

2.1 Pre-processing

We start with label encoding, converting the ‘humour’ category to ‘fake’ and mapping ‘fake’ to 0 and ‘real’ to 1, thus binarising our target variable for the binary classification task. This step is pivotal as it simplifies the output space for our algorithms, aligning with best practices in binary classification tasks.

The inclusion of ‘@mentions’ in our analysis is based on the understanding that fake news often uses tagging to boost its visibility. By examining the interactions in each post, we aim to identify patterns that indicate the authenticity of the content. This approach is supported by research showing that mentions and replies in tweets, particularly in the news, strategically enhance visibility and engagement. Studies have demonstrated that tweets with mentions or replies receive more attention, suggesting a link between these elements and tweet engagement. Therefore, analysing the frequency and context of ‘@mentions’ could help detect potential fake news by artificially revealing efforts to increase a post’s reach [6].

Moving on to processing the tweet, starting with lowercasing, which forms the foundation of our text normalisation process. This simple yet effective technique ensures consistency across the dataset, irrespective of how users originally formatted their text. By converting all text to lowercase, we eliminate any discrepancies occurring from case differences, thus simplifying the dataset and enhancing the accuracy of feature representation.

Next, potential data redundancy is addressed by removing retweets. Retweets can lead to duplicative content, potentially leading the model to false readings. We ensure the model’s learning is based on original and unique content by eliminating retweets, typically marked by ‘RT’ or similar indicators.

Another critical aspect is the removal of URLs, which are often irrelevant to the text’s sentiment or factual content. Given their random nature and lack of correlation with news veracity, URLs are removed to streamline the feature space, thereby preventing unnecessary inflation of the model’s input dimensions without any significant analytical gain.

The removal of special characters and numbers further refines the text. These elements are removed as they often don’t contribute to the text’s meaning. This step simplifies the textual data, allowing the model to focus on meaningful content signifying news genuineness.

Removing stopwords is employed to reduce dimensionality and enhance focus on relevant content. This involves filtering out common words that appear frequently across texts but generally do not carry significant meaning or distinguish between real and fake news.

Next, lemmatisation plays a crucial role in maintaining the contextual meaning of the text. This process normalises various word forms to their base form, ensuring that the model does not treat them as separate features. This step is critical in capturing the essence of the text more accurately and is preferred over stemming due to its context-sensitive approach.

A step in our preprocessing pipeline was the removal of columns such as ‘tweetId’, ‘userId’, ‘username’, and ‘imageId(s)’. This action was necessary to avoid model bias based on arbitrary identifiers. In addition, this data would not help determine if a post is ‘real’ or ‘fake’.

To measure the complexity of the text, we incorporated readability assessments, specifically the Flesch Reading Ease and Gunning Fog Index. These metrics provide valuable insights into the textual complexity, aiding in distinguishing between real and fake posts. This method is rooted in research findings that fake news uses more straightforward language, fewer technical terms, and more redundancy. These characteristics make fake news more accessible and less demanding regarding the reader’s education level [4]. By leveraging these readability metrics, we aimed to exploit these linguistic differences, enhancing our model’s ability to differentiate between ‘real’ and ‘fake’ news.

Another feature we calculated was the unique word count in each post. This is significant because ‘real’ news usually employs a broader vocabulary, while ‘fake’ news often leans on repetitive terms. This linguistic distinction is a subtle yet important attribute of ‘real’ news.

We also performed sentiment analysis on each post, calculating the overall sentiment polarity. The hypothesis underpinning this analysis was that ‘real’ news might exhibit different emotional tones compared to ‘fake’ news. This difference in sentiment can be pivotal in distinguishing between the two, as ‘fake’ news often aims to evoke stronger emotional reactions to virality.

Finally, we extracted features such as the day of the week and the hour of the day from the timestamps. This step was based on the premise that the distribution of ‘real’ and ‘fake’ posts might follow different patterns. ‘Real’ news might be distributed more evenly, while fake news could show spikes during certain times.

2.2 Feature Selection and Dimensionality Reduction

The feature selection and dimensionality reduction process was vital in optimising model performance, particularly in balancing the trade-off between complexity and overfitting. The selected features ‘polarity’, ‘gunning fog’, and ‘sentiment’ emerged as the most effective in differentiating real from fake news, as evidenced by extensive analysis and experimentation on the dataset. These features were theoretically aligned with the characteristics of fake news. However, it also demonstrated the best results regarding model accuracy and F1 scores for this specific dataset. In addition to their theoretical backing, each feature was validated through iterative testing and refinement. This practical approach allowed a nuanced understanding of how each feature contributed to the model’s overall performance. The model’s reliance on these selected features was further justified by their consistent performance across various configurations and parameter settings.

Acknowledging that the chosen features were part of a broader strategy to manage the dataset’s dimensionality is crucial. By focusing on a limited but highly effective set of features, the model could avoid the pitfalls of high-dimensional data, such as overfitting and increased computational demand.

The approach to dimensionality reduction was carefully selecting features based on their theoretical relevance and practical performance and limiting the number of features to maintain model simplicity and interpretability. This method ensured that the model was equipped to capture the essential characteristics of the data while remaining robust and generalisable.

Moving on, using TfidfVectorizer in the data pipeline demonstrates a reasonable approach to feature engineering. This tool transforms textual content into a matrix of TF-IDF features, which is crucial for quantitatively analysing text within the model. More than a mere transformation tool, TF-IDF is pivotal in highlighting unique words within documents. By reducing the impact of common yet less informative words, TfidfVectorizer sharpens the model’s focus on distinct and meaningful terms in the dataset, ensuring analytical precision without unnecessarily expanding data dimensionality. This straightforward encoding of TF-IDF not only aids in forming the basis for complex algorithms and query retrieval systems but also enhances the relevance and accuracy of the model in discerning the authenticity of news content [7].

Finally, to address the imbalance of ‘fake’ to ‘real’ labels in the dataset, integrating SMOTE (Synthetic Minority Over-sampling Technique) into the data pipeline was a strategic decision. This imbalance, marked by a higher prevalence of ‘fake’ labels, posed a risk of biasing the classifier against the ‘real’ class. By synthesising new examples in the minority class, SMOTE effectively balances the dataset, fostering a more equitable learning environment for the model. This choice is not just a mere procedural step but a crucial part of the method to ensure that the model’s learning is comprehensive and unbiased. The employment of SMOTE reflects a deliberate effort to overcome one of the inherent challenges of the dataset, aligning with established practices in handling imbalanced data [1].

2.3 Machine Learning Algorithm

The choice of machine learning algorithms was a crucial component of the pipeline. Logistic Regression and Random Forest were selected for their distinct characteristics and proven effectiveness in binary classification tasks.

Logistic regression was chosen for this task for its applicability as a supervised machine learning algorithm for classification, mainly when dealing with binary data. The strength of logistic regression lies in its use of the logistic/sigmoid functions to model the probability of a specific class or event. This characteristic makes it especially suitable for scenarios with binary outcomes

[3]. The algorithm assigns a score between 0 and 1 through the sigmoid function, allowing for an interpretation of the likelihood of a tweet being ‘real’ or ‘fake’.

This approach is consistent with the dataset’s characteristics and the nature of the classification problem at hand. The selection of logistic regression is further justified by its widespread use in text classification tasks, where the aim is often to categorise textual data into predefined classes [5]. This algorithm excels in scenarios where the relationship between independent variables and the binary outcome is approximately linear, making it a suitable choice for text classification tasks. Logistic regression’s simplicity and interpretability make it a staple in textual data analysis areas.

The Random Forest algorithm, recognised as one of the most influential classification algorithms in machine learning, was selected for its robustness and versatility. Its unique approach of combining multiple decision trees significantly increases its predictive accuracy and reduces the risk of overfitting. Each tree in the Random Forest operates independently, and their collective output is used to arrive at the final decision, thereby enhancing the overall reliability of the model [3].

This algorithm’s diversity in handling different types of input data and its simplicity in implementation make it highly suitable for complex tasks like fake news detection. Random Forest’s strength lies in its uncorrelated nature; the more diverse the trees, the higher the accuracy of the model. This diversity is particularly beneficial in dealing with datasets with a wide range of features, ensuring that any single feature or pattern does not overly influence the model.

3 Evaluation

The F1 score is a vital metric in evaluating the balance between precision and recall, especially important in datasets with class imbalances like ours. It is calculated as the harmonic mean of precision (the proportion of true positive predictions among all positive predictions) and recall (the proportion of true positive predictions among all actual positives). The formula for the F1 score is:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.1 Initial Model Performance

Initially, the evaluation phase focused on assessing Logistic Regression and Random Forest models using their default parameters. The results on the training dataset were as follows:

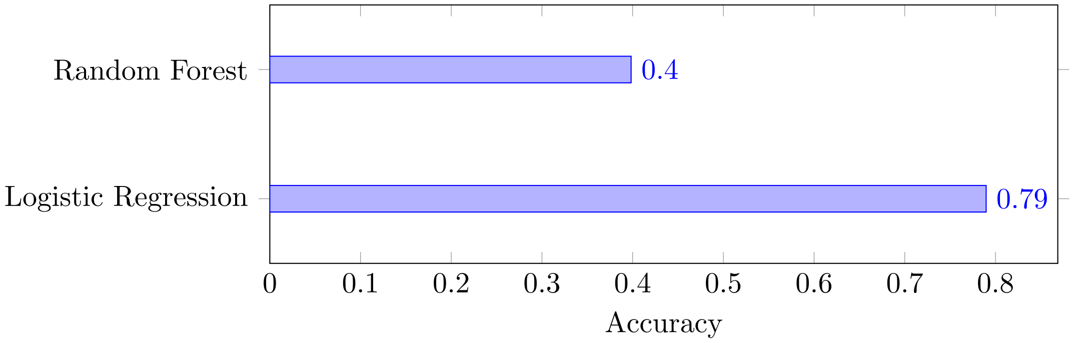

Figure 3
Initial Accuracy of Logistic Regression and Random Forest on Test Data

These initial scores were crucial in understanding the models’ effectiveness in distinguishing between ‘real’ and ‘fake’ news posts. While Logistic Regression showed a relatively high F1 score, indicating a better balance between precision and recall, the Random Forest model's performance was notably lower, suggesting challenges in dealing with the dataset's complexities.

3.2 Hyperparameter Tuning: Bayesian Optimization Approach

Bayesian optimisation was employed to optimise the hyperparameters of both Logistic Regression and Random Forest models. This method stands out for its informed, iterative search strategy, where learnings from each iteration inform subsequent ones. Unlike Random Search, Bayesian Optimization systematically selects hyperparameter combinations based on a surrogate model, commonly a Gaussian Process. By focusing on maximising the F1 score, the models were fine-tuned to achieve an optimal balance between precision and recall, enhancing their performance in classifying fake news content [2].

3.2.1 Logistic Regression Hyperparameter Tuning

For Logistic Regression, Bayesian optimisation was employed using Hyperopt. This approach allowed for an efficient exploration of the hyperparameter space. The key parameters adjusted included regularisation strength ('C'), maximum iterations ('max iter'), solver type, and class weighting. The optimisation process was iteratively performed with ten evaluations, each time refining the parameter choices based on the F1 score. In addition, Stratified K-Fold Cross-Validation in the tuning ensured a balanced representation of classes, thus providing a more accurate assessment of the model's performance.
Resulting Paramters:

- **Best 'C' value: 1**
- **Best 'max iter': 5000**
- **Best solver: lbfgs**

3.2.2 Random Forest Hyperparameter Tuning

Random Forest's tuning involved adjusting parameters like the number of estimators ('n estimators'), tree depth ('max depth'), the minimum samples for a split ('min samples split'), and the minimum samples for a leaf ('min samples leaf'). The tuning was conducted through Bayesian optimisation over ten evaluations, focusing on increasing the model's accuracy and reducing the likelihood of overfitting.
Resulting Paramters:

- **Best 'n estimators': 300**
- **Best 'max depth': None**
- **Best 'min samples split': 5**
- **Best 'min samples leaf': 4**

3.3 After Hyperparameter Tuning

3.3.1 Post-Tuning Performance Metrics

Surprisingly, post-tuning, the F1 scores for both the Logistic Regression and Random Forest models on the test dataset remained consistent with their pre-tuning performance. The F1 Score for Logistic Regression held steady at 0.7897, and Random Forest's F1 Score remained at 0.3990. This outcome indicates that this dataset's initial model parameters were already near optimal. In the domain of fake news detection, it is widely recognised that obtaining high F1 scores is challenging. However, another project working with a textual dataset noted similar challenges, emphasising the difficulty in achieving high accuracy due to the dataset's complexity and imbalance [8].

Model	**TP**	**TN**	**FP**	**FN**	**AUC**	**F1 Score**
Logistic Regression	946	2093	453	263	0.79	0.79
Random Forest	20	2319	207	1189	0.24	0.40

Figure 4: Post-tuning results

Note:

- **TP (True Positives)**: Number of positive instances correctly classified.
- **TN (True Negatives)**: Number of negative instances correctly classified.
- **FP (False Positives)**: Number of negative instances incorrectly classified as positive.
- **FN (False Negatives)**: Number of positive instances incorrectly classified as negative.
- **AUC (Area Under Curve)**: Represents the model's ability to discriminate between positive and negative classes.
- **F1 Score**: Harmonic mean of precision and recall, providing a balance between them.

In the post-tuning performance comparison, Logistic Regression and Random Forest models exhibited notable differences in classifying 'real' and 'fake' posts. Logistic regression proved more proficient at identifying 'fake' posts, with 946 true positives and only 263 false negatives, while Random Forest struggled, identifying only 20 true positives and a significantly higher 1189 false negatives. In classifying 'real' posts, Logistic Regression had 2093 true negatives against 453 false positives, whereas Random Forest showed a stronger tendency towards 'real' classification with 2319 true negatives but fewer false positives at 207. Logistic Regression's AUC of 0.79 indicates a balanced sensitivity and specificity, unlike Random Forest's lower AUC of 0.24. The F1 scores further emphasise this trend: Logistic Regression achieved a balanced score of 0.79, while Random Forest lagged at 0.40, primarily due to lower recall. These results highlight Logistic Regression's overall balanced efficiency in classification compared to Random Forest's tendency to favour 'real' post-classification.

3.3.2 Insights and Analysis

The F1 scores post-hyperparameter tuning, at 0.7897 for Logistic Regression and 0.3990 for Random Forest, raise important questions about the interaction between machine learning models, their parameters, and the datasets they are applied to. This outcome separates from the common expectation where tuning is anticipated to yield significant improvements in model performance.

The consistency in F1 scores suggests two possible scenarios. Initial model efficiency, where the default parameters for both models might have been well-suited for this dataset, leaving little room for improvement through further tuning. Performance limitations, where the models, particularly the Random Forest, might have reached their performance ceiling with this specific text classification task. The unchanged results post-tuning also shed light on the difficulties of hyperparameter optimisation. While Bayesian Optimization is a powerful tool, this scenario illustrates that its effectiveness can be bounded by the nature of the dataset and the inherent capabilities of the chosen algorithms.

This study's limited impact of hyperparameter tuning could also be down to computational limitations. A constrained exploration of the hyperparameter space due to computational resources may have prevented a more thorough and worthwhile tuning process. This highlights the practical challenges in machine learning, where ideal computational power and resource conditions are not always available.

The consistent F1 scores following hyperparameter tuning also prompt a reflection on the dataset's intrinsic characteristics. It raises the possibility that the nature of this specific dataset may need to be more beneficial to make significant improvements through hyperparameter optimisation. This could be attributed to various factors inherent to the dataset, such as the complexity of the text and the subtleties in the distinctions between 'real' and 'fake' news. This insight emphasises the crucial role that dataset characteristics play in the effectiveness of machine learning models and their optimisation processes.

4 Conclusion

In conclusion, this study aimed to develop a machine-learning pipeline for distinguishing between real and fake news posts on social media. The approach involved preprocessing the textual data, selecting key features, and employing Logistic Regression and Random Forest models. Initial evaluations and subsequent hyperparameter tuning highlighted exciting insights about the interaction between algorithms and the dataset's characteristics.

The preprocessing stage focused on normalising the text and extracting features that could distinguish between real and fake news. The choice of Logistic Regression and Random Forest was motivated by their suitability for binary classification tasks and differing approaches to handling complex datasets.

The evaluation phase revealed that the Logistic Regression model consistently outperforms the Random Forest model. However, the most intriguing finding was that hyperparameter tuning did not significantly enhance the models' performance, suggesting either the initial parameters were near-optimal or the models reached their performance limits, given the dataset's complexity.

Several insights were gained from these findings. Firstly, the dataset's characteristics are crucial in determining model performance. Secondly, while hyperparameter tuning is often beneficial, its impact can be limited by the nature of the data and the inherent capabilities of the models used. For future research, it would be interesting to explore other machine learning algorithms or ensemble methods that might offer improved performance for this specific task. Additionally, incorporating multimodal data such as images or videos could provide a more comprehensive approach to fake news detection. The study also underlines the importance of considering computational resources in the context of machine learning, as limitations in this area can impact the feasibility and effectiveness of specific techniques. If more computational resources were available, we could see how much of an effect hyperparameter tuning would have over a greater space.

References

[1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[2] Enas Elgeldawi, Awny Sayed, Ahmed R Galal, and Alaa M Zaki. Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. In *Informatics*, volume 8, page 79. MDPI, 2021.

[3] Sayar Ul Hassan, Jameel Ahamed, and Khaleel Ahmad. Analytics of machine learning-based algorithms for text classification. *Sustainable Operations and Computers*, 3:238–248, 2022.

[4] Benjamin Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 759–766, 2017.

[5] Bipin Nair B J, S. Yadhukrishnan, and Manish. A. A comparative study on document images classification using logistic regression and multiple linear regressions. In *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, pages 1096–1104, 2023.

[6] Claudia Orellana-Rodriguez and Mark T Keane. Attention to news and its dissemination on twitter: A survey. *Computer Science Review*, 29:74–94, 2018.

[7] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[8] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.