

# Big Data Analysis and Project

## Assignment 1 – Part D

### To analyze the impact of the figure of speech, sentiment, and social media on the detection and propagation of fake news over social media.

This report investigates the effects of social media, language, etc. on how fake news spreads online. To understand this, we used a variety of data sources, including news articles and tweets. We developed computer models that could distinguish between true and false news, and the results indicated that these models could be useful in the fight against false information on social media. This report summarizes our project tasks, provides conclusions, and makes recommendations for how to improve the situation and carry out further research.

#### Part 1- Restatement and Summary

This study explores the effects of social media, sentiment, and figures of speech on the identification and spread of false information [5]. By analyzing news articles and tweets, the study addresses key questions about the capability of machine learning algorithms in distinguishing genuine from fake news. For identifying such fake news, we improved machine learning models like the Naive Bayes Classifier and K-Nearest Neighbours. The investigation also investigated the connection between the spread of fake news and ironic or sarcastic tweets [6]. Additionally, to improve performance, data preprocessing methods like word vectorization and cleaning were used, along with Kernel Density Estimate and Hyperparameter tuning.

```
train_data = pd.read_csv("D:\\UoA\\Trimester 2\\Big Data Analysis and Project\\train.csv")
test_data = pd.read_csv("D:\\UoA\\Trimester 2\\Big Data Analysis and Project\\test.csv")

train_data.head()
```

[18] ✓ 0.1s

	tweets	class
0	Be aware dirty step to get money #staylight ...	figurative
1	#sarcasm for #people who don't understand #diy...	figurative
2	@lminworkJeremy @medsingle #DailyMail readers ...	figurative
3	@wilw Why do I get the feeling you like games?...	figurative
4	-@TeacherArthurG @rweingarten You probably jus...	figurative

```
test_data.head()
```

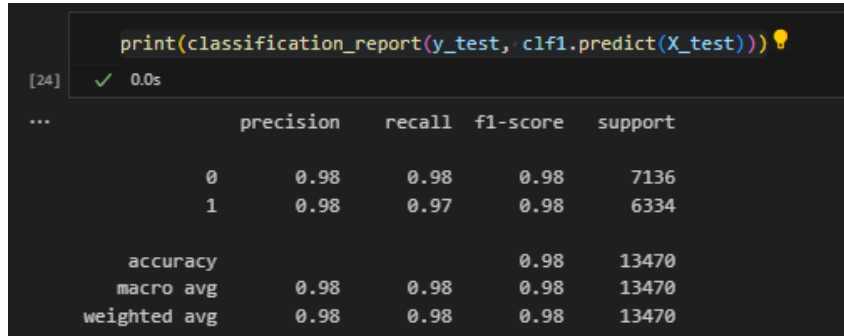
[17] ✓ 0.0s

	tweets	class
0	no one ever predicted this was going to happen...	figurative
1	@Stooshie its as closely related as Andrews or...	figurative
2	I find it ironic when Vegans say they love foo...	figurative
3	Quick rt that throwing money vine I've not see...	figurative
4	yep, keep adding me to your #devops lists....	figurative

Figure 1- A brief overview of the dataset used in the project. The figure contains both the training and the testing dataset.[1][2]

## Part 2- Analysis and Visualization

We examined the findings from the initial models and their subsequent refinement during the analysis phase. We used Multinomial Naive Bayes (NB) and K-Nearest Neighbours (KNN) algorithms in the initial model to categorize news articles based on titles. The initial model gave us the results as shown in the figure below.



```
print(classification_report(y_test, clf1.predict(X_test)))
```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	7136
1	0.98	0.97	0.98	6334
accuracy			0.98	13470
macro avg	0.98	0.98	0.98	13470
weighted avg	0.98	0.98	0.98	13470

Figure 2- The results of our initial implementation for the Multinomial Naïve Baye’s Classifier **before the refinement**.

The initial models offered insightful information about how well the classifiers performed. We further investigated the models' hyperparameters and used a grid search with cross-validation to identify the ideal alpha value for Multinomial NB to improve our strategy. This aimed to improve the models' ability to distinguish between real and fake news. With the help of this model, we were able to better distinguish the tweets from one another based on their sarcastic tone [3]. This can be depicted by the figures below:

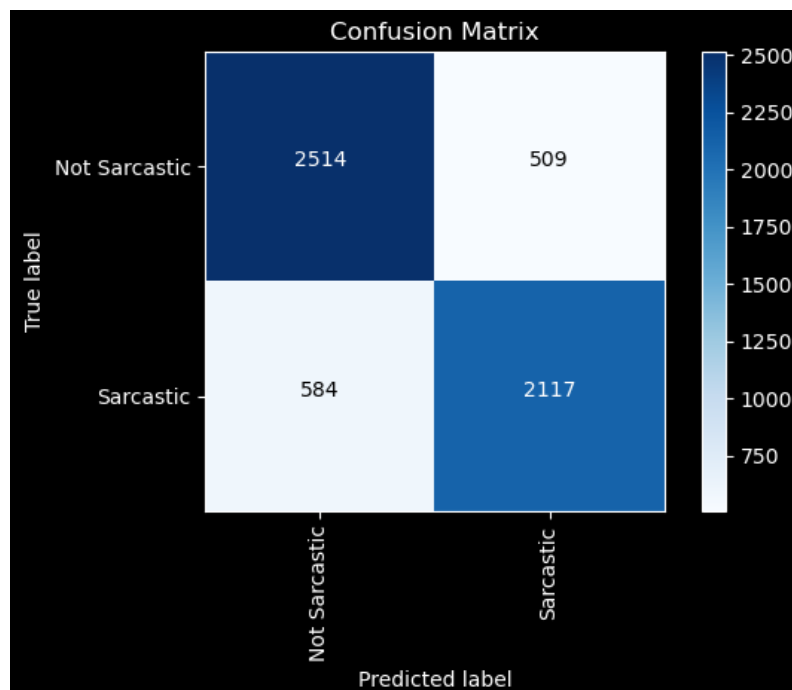


Figure 3- A confusion matrix generated from our initial model implementation, which gives us information about the tweets data, based upon sarcasm.

After our model was refined, the updated confusion matrix can be viewed below:

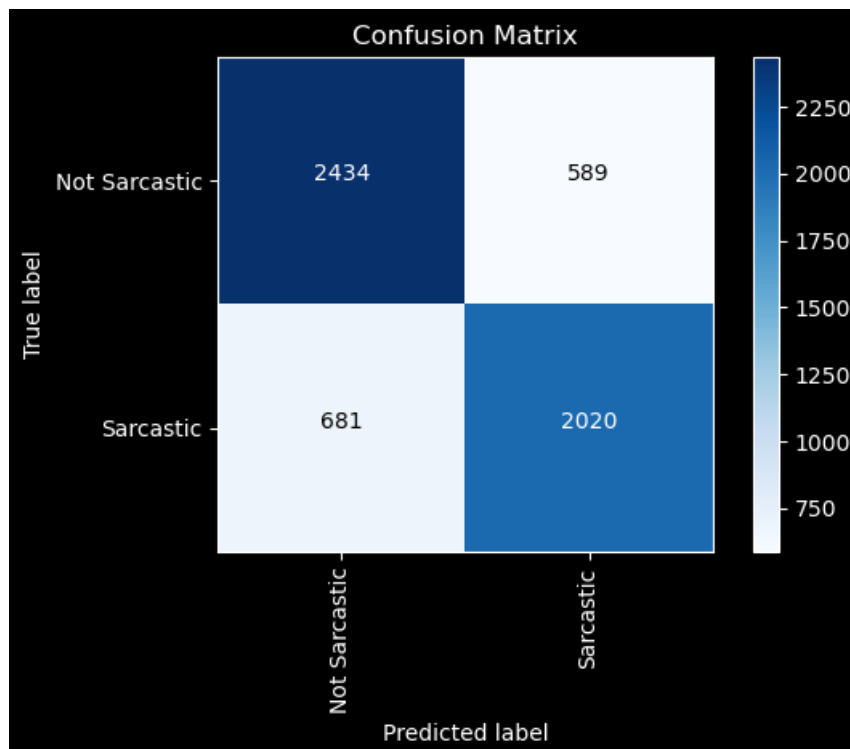


Figure 4- The updated confusion matrix which is derived *after* our model has been refined. It can now be seen the classification has been more precise and accurate.

We found significant performance gains for the models after refinement. Higher precision and recall scores from the improved models demonstrated an improvement in their ability to correctly classify news articles. Using this improvement, and our updated model, we were able to get a more precise distinction between fake and real news.

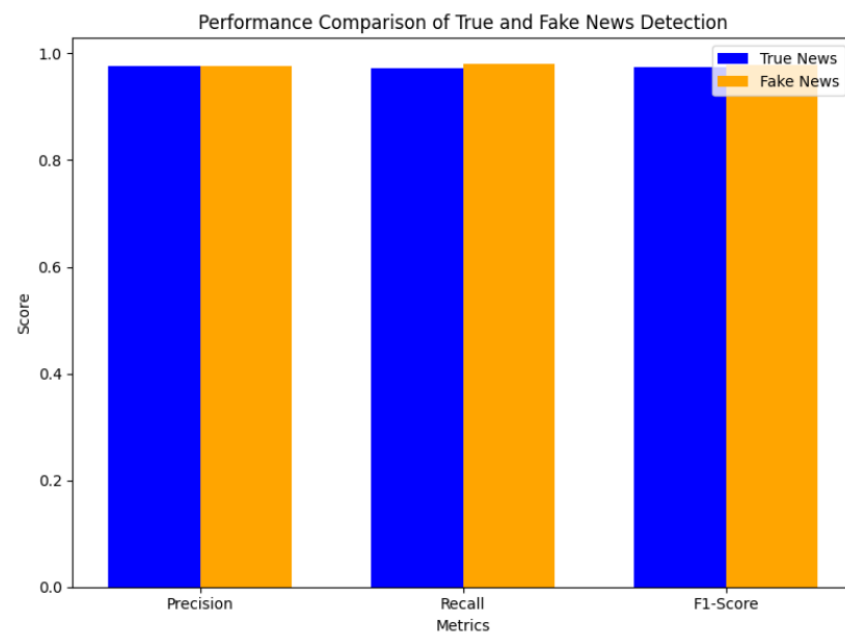


Figure 5- The grouped bar chart that compares the precision, recall, and F1-score of our model for detecting true and fake news, *after refinement*.

We made sure that our analysis was founded on the most accurate and efficient classification approach by evaluating and improving our models iteratively. This process improved the

predictive ability of the models while also highlighting how crucial continuous improvement is to get accurate results. [7]

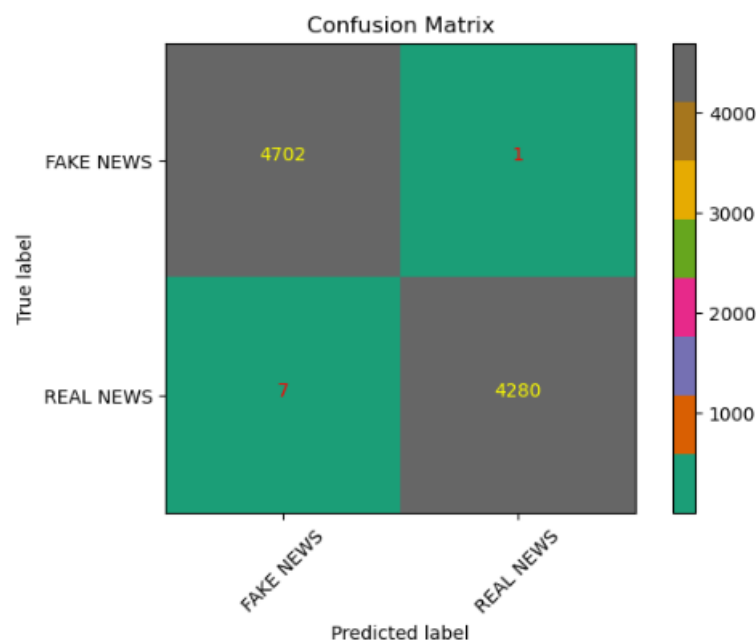


Figure 6- Final confusion matrix of our model (after hyperparameter tuning and other refinement tasks), distinguishing fake and real news.

### **Part 3- Improvement of the situation**

The used models have been improved using methods like feature scaling and hyperparameter tuning, which have improved the classification accuracy. We can experiment with ensemble techniques, explore advanced NLP methods, and adjust model parameters to further improve our models. This iterative refinement process can enhance our system's overall functionality and help us detect fake news with greater accuracy.

The actions listed below can be used to implement these improvements:

1. *Feature engineering*: We can use sentiment analysis libraries like VADER or TextBlob.
2. *Ensemble Methods*: We can use techniques like bagging or boosting to combine multiple models and use sci-kit-learn libraries to quickly implement ensemble classifiers.
3. *Domain-Specific Features*: We can develop a feature extraction process that takes source credibility metrics and article topics into account.
4. *Advance NLP Models*: Utilize advanced NLP models, such as BERT, for improved word embeddings.
5. *Regularization Techniques*: Use regularization techniques to control overfitting, such as dropout, L1/L2 regularization, and cross-validation.

These actions can be taken to improve the system's ability to recognize and stop the spread of fake news on social media.

#### Part 4- Conclusion and Future Work

By examining figures of speech, sentiment, and the impact of social media, our project successfully addressed the detection and spread of fake news. We improved the accuracy of separating fake from real news by improving models. The study emphasizes the capability of machine learning for accurate fake news detection and provides practical advice for battling false information on social media platforms.

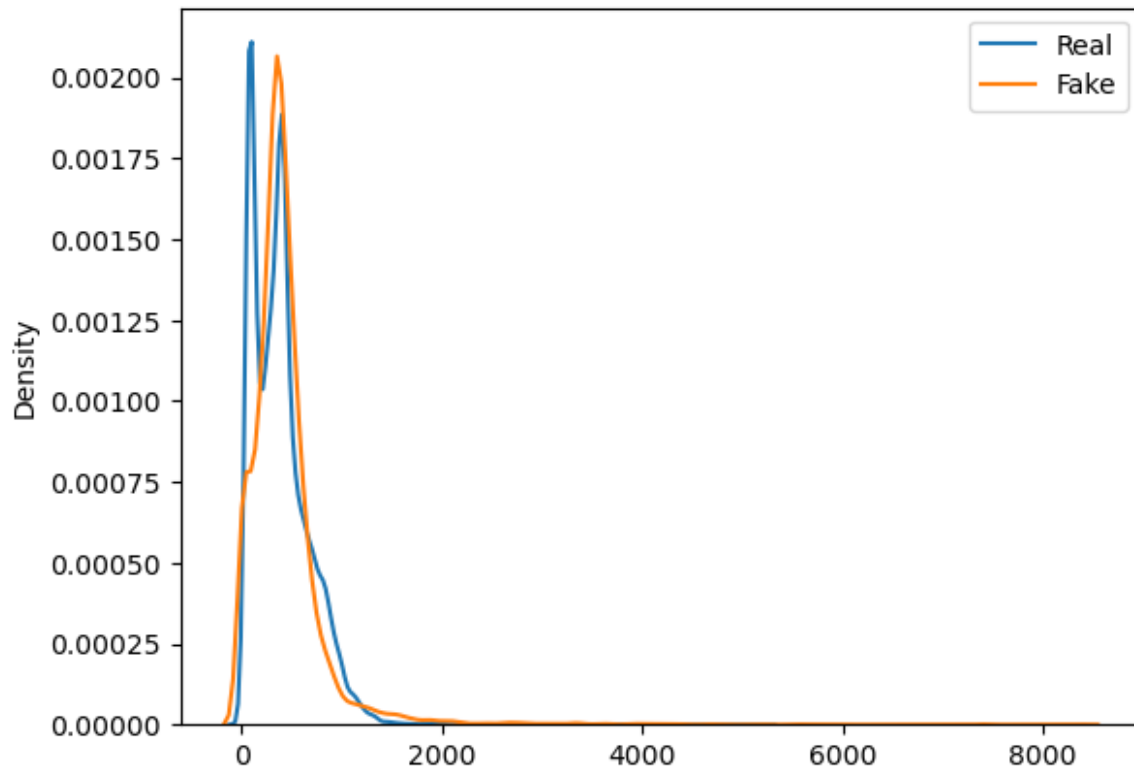


Figure 7-One of the outcomes we came across in our project, regarding article lengths, is that Fake articles don't have that spike around low lengths that Real ones do, using KDE Plot.[1][4]

Future work might incorporate more sophisticated NLP methods, investigate bigger datasets, and investigate how sentiment analysis affects the accuracy of fake news detection. We can also make use of multiple improvement techniques and tasks we can carry on working for even better efficiency. This project lays the groundwork for additional research, which will eventually contribute to more effective fake news detection techniques, fostering an informed and trustworthy digital environment.

#### References:

- 1- [www.kaggle.com](https://www.kaggle.com). (n.d.). *Fake and real news dataset*. [online] Available at: <https://www.kaggle.com/datasets/clmentbisailon/fake-and-real-news-dataset>
- 2- Quantitative Finance & Algo Trading Blog by QuantInsti. (2022). *Natural Language Processing in Python Using spaCy*. [online] Available at: <https://blog.quantinsti.com/spacy-python/> [Accessed 4 Jul. 2023].

- 3- [www.kaggle.com](https://www.kaggle.com/datasets/nikhiljohnk/tweets-with-sarcasm-and-irony). (n.d.). *Tweets with Sarcasm and Irony*. [online] Available at: <https://www.kaggle.com/datasets/nikhiljohnk/tweets-with-sarcasm-and-irony>.
- 4- Pydata.org. (2012). *seaborn.kdeplot — seaborn 0.9.0 documentation*. [online] Available at: <https://seaborn.pydata.org/generated/seaborn.kdeplot.html>.
- 5- Bostonia. (n.d.). *Fake News Influences Real News*. [online] Available at: <https://www.bu.edu/bostonia/2017/fake-news-influences-real-news/> [Accessed 25 July 2023].
- 6- (Singh, 2019) [www.kaggle.com](https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection). (n.d.). *News Headlines Dataset for Sarcasm Detection*. [online] Available at: <https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection>.
- 7- Rosebrock, A. (2021). *Introduction to hyperparameter tuning with sci-kit-learn and Python*. [online] PyImageSearch. Available at: <https://pyimagesearch.com/2021/05/17/introduction-to-hyperparameter-tuning-with-scikit-learn-and-python/>.