Fake News Detection using LSTM Neural Network

*AML 3104 Neural Networks and Deep Learning*

*Presented by*

Swathi Gurijala (C0790294)

Varadharajan Kalyanaraman (C0793756)

Vignesh Kumar Murugananthan (C0793760)

Under the Guidance

Of

Prof. Mohammad Islam

**Table of Contents**

[Abstract 3](#_Toc90218600)

[1. Introduction 4](#_Toc90218601)

[2. Problem Definition 4](#_Toc90218602)

[3. Methodology 4](#_Toc90218603)

[4. Implementation 5](#_Toc90218604)

[5. Exploratory Data Analysis 5](#_Toc90218605)

[6. Data Pre-Processing 7](#_Toc90218606)

[7. Data Modeling 7](#_Toc90218607)

[8. Model Evaluation 9](#_Toc90218608)

[9. Results 9](#_Toc90218609)

[References 11](#_Toc90218610)

Abstract

In the early days, people relied on specific news sources for their trusted information for years. With technological advancements, the internet has become cheaper now. And people of different regions and age groups have access to it. So, the news that an average individual receives is from popular social media platforms like Facebook, Twitter, YouTube, and blogs. There is a high possibility that the data could be fake and unreliable. People in these platforms can also misuse it by explicitly crafting the wrong information and spreading false perceptions all around. The government and private agencies are also working together to reduce the impact of fake news on people.

This project will create a machine learning algorithm that tells if the given information is genuine or fake. The machine learning algorithms proposed are NVB, Decision Trees, Passive-aggressive algorithms, and recurrent neural networks named Long Short-Term Memory (LSTM).

Keywords- Natural Language Processing, Machine Learning, Fake news detection, TRP-Television Rating Point, API- Application Programming Interface

1. Introduction

“Innocent until proven guilty” – The presumption of innocence. Before the court verdict, social media platforms boost the wave of judgment and opinions. These become worst when it is too late to prevent disaster. Ultimate, injustice! All because of the “Fake News,” defined as fabricated news in short misinformation. It has been identified as a global threat.

We must rescue ourselves from them, and for that, we need to find ways to prevent fake news. So, people will be well-informed, counteract misleads, and be aware of the fact. Stopping the popularity of social media platforms and TRP eaters can help business boosters and society. Instead, Google attempts to stop providing its ad services to fake news websites. We need to build a system that can identify and detect Fake news after filtering through non-systematic fake news detectors of a few social media platforms like Facebook. So, we can persist the trust of people towards the media mass.

Facebook, Twitter, or any media platforms have their fake news detector, but it can process only that platform’s related feeds. What if open to all concepts deployed so people can paste text over the provided box and get the truthfulness of the news.

1. Problem Definition

Detection of Fake News involves the identification of the form of the news. Whether the intentionally formed? Waving false stories? Purposely threat and violence flame? All these defers from the facts. The primary approach of this project is to get enough text with an identified label then train the machine learning model. Evaluate the best suitable model and choose it for the backend heart. Export it into the pickle file, implement the user-friendly front-end. End-user can pass the text, which can be fed to the pickled model, and the classification result fetch to the front-end and show the result to the end-user.

1. Methodology

Our project has mainly three components which are –

* User Interface [Front-end]
* Machine Learning Model [Backend]

**User Interface**

Flask is a framework that provides libraries to build very lightweight applications using python.  Despite that, Mario lets you develop the publication very quickly, and you can also extend it since it is a microphone work that does not include ORM, it has many excellent features that include URL routing and template engine.

**Machine Learning Model**

Fake news detection is a tedious task to perform. Making a machine understand how to classify whether a news source is fake or true by understanding previous trends and patterns in the dataset and classifying new data points is the essential part of our model. To ensure our model predicts accurate results, we had to implement and test out various machine learning algorithms and choose the best one. There are many steps to be followed in finalizing our model, and they are discussed further.

1. Implementation

As described earlier about the two components of the project, we have started from the machine learning model.

**Data Collection**

Data collection is crucial for any Machine learning project; it is vital to choose the best relevant, trustworthy, and suitable subject source for the pipeline. We have selected the most trusted ‘Kaggle’ portal for the data source and included the diversity merged the ‘Flair’ provided dataset not ended up with these we also gathered the data from the Twitter [API]. These three sources have finalized our dataset.

As our data sources are different, the structure of the data features has also differed. It was an essential step to create uniformity among them. Started to process individual datasets and mold them into the same structure, then merged them. Here, the Kaggle dataset was in two different files, labeled True as a separate file while False was indifferent, added a new column as the label, and merged that. On the other hand, the Flair dataset had categorical labels and converted them numerically. The main issue was with Twitter posts, as they had to be labeled by checking the manual authenticity – surfing through web portals. After completing that final dataset, have more than 50K records with two features – Text and the target Label.

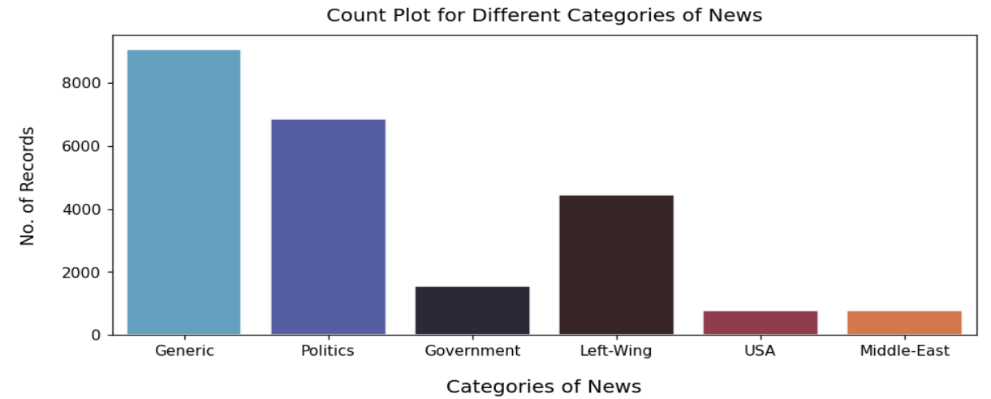
* Text – The title of news + text of the news
* Label – Fake/True

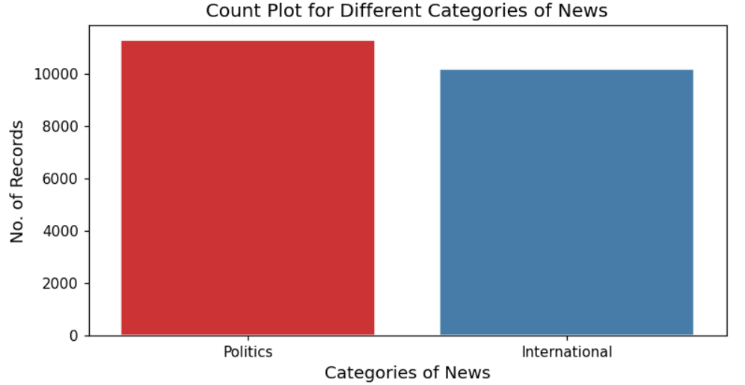
1. Exploratory Data Analysis

Exploratory Data Analysis is a process to analyze the dataset and understand the hidden pattern, trends, and dataset’s main characteristics, often using visual methods. Visualizing data aims to understand and interpret information out of the raw dataset. Analyzing data using from scratch functions is time-consuming and a big hassle.

In Data visualization, the most crucial package everyone uses is matplotlib. It provides us granular control on our plots and helps us tweak every pixel of them. Most of the visualization packages built now call matplotlib API in their backend. One of the packages that are built on such a structure is seaborn. It is a comparatively better package than matplotlib in terms of fewer lines of code to plot various charts. We also have many other packages like plotly, cufflinks, and dash – highly interactive and dynamic visualization packages.

In our dataset, we have identified that there are different categories of news in the two datasets. We have visualized these data as follows.



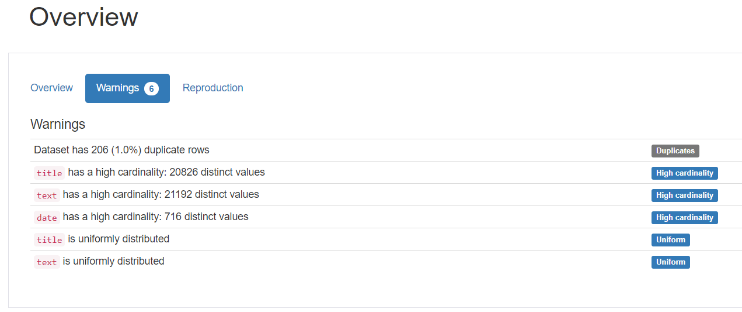
**Figure 1: Different Categories of news of Fake dataset**.

**Figure 2: Different Categories of news of Real dataset.**

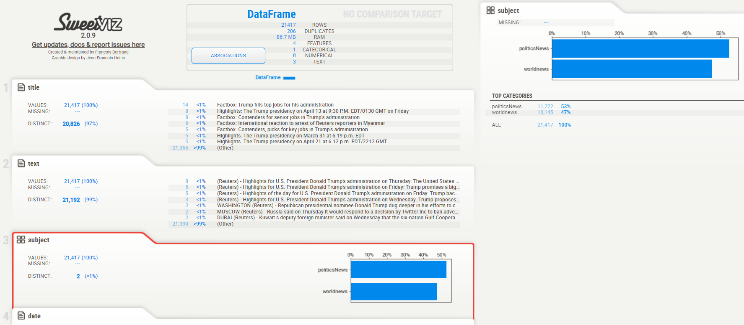
As we can see from the above images, we understand that in both True and Fake datasets, we have different news categories.

**Profiling Reports**

Profiling Reports are a modern way to interpret data without writing many lines of code and using the packages efficiently to understand and interpret the information about the dataset from the profiling reports. Profiling Reports can be generated using various packages like:

We have used both pandas\_profiling and sweetviz on our datasets. A few of the screenshots are mentioned below. As they are interactive, we can have an entire dataset description.

**Figure 3: Pandas Profiling.**

**Figure 4: Sweetviz**

1. Data Pre-Processing

Once we gain insight into our datasets, we need to go further in building an efficient machine learning model.

**Dataset Manipulation**

As per our project need, we had three different data sources, and to maintain a unique dataset, we had to merge all three data sources into one. After merging, we had only two columns, namely text, and label, where text is the news article and label is the classification indicator whether it is True or False.

As three different data sources are merged, there are high chances of duplicate values that are negated by removing duplicates from our merged dataset.

**Text Cleaning**

We have cleaned our text using RegEx’s sequence- regular expressions: the cleaned text has more than 45K records.

| **Remove** | **Description** |
| --- | --- |
| Punctuations | String. punctuations a |
| White spaces | Removed extra white spaces using RegEx |
| Case | Converting to lowercase |
| StopWords | Using nltk. corpus Stop words |

**String import**

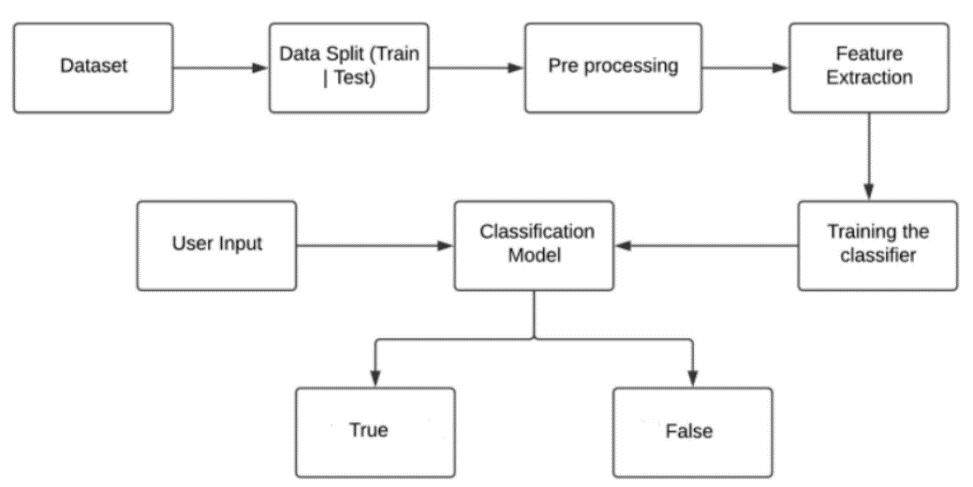
It is essential to do the Stemming here, as Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or the roots of words known as a lemma.

We have used One of the most popular stemming algorithms: Porter stemmer, which has been around since 1979. We have included the function stemming in our cleaning function itself, which will stem after the cleaning.

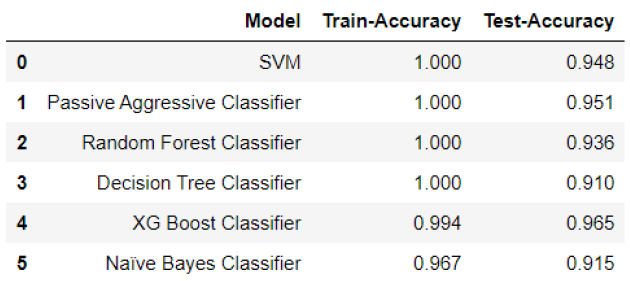
1. Data Modeling

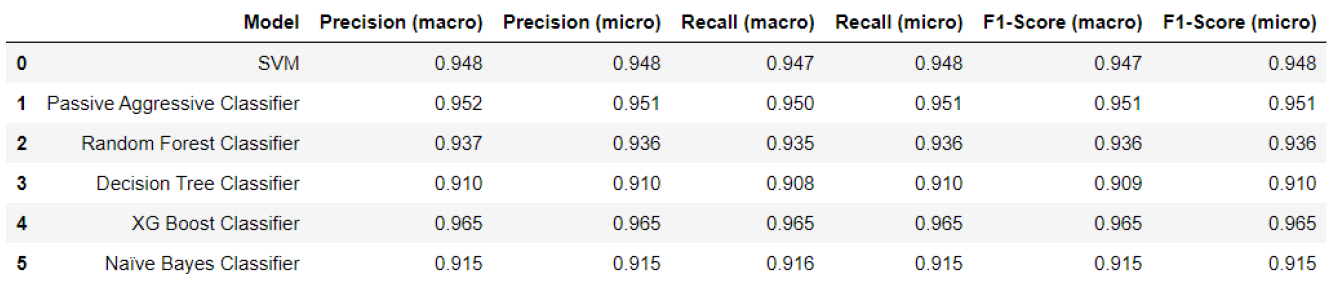
Data modeling is the most important part of the machine learning model. This is the stage where we decide which machine learning algorithm is best suited for our problem statement. After thorough research, we have decided to go with the following machine learning algorithms.

* Support Vector Machine
* Passive Aggressive
* Random Forest
* Decision Tree
* XGBoost
* Naïve Bayes
* Long Short-Term Memory



**Figure 5: Model Flow.**

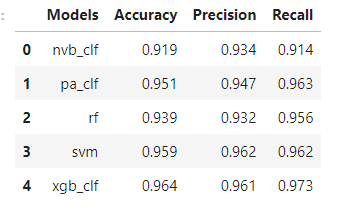


**Figure 6: Test-Train Accuracy**

**Figure 7: Model Evaluation Metrics.**

**Performance Improvement**

After analyzing the accuracy, we found that a few of the models were overfitting, which can be improved by Hyper-parameter tuning. Using suitable parameters of each model, we have implemented GridSearchCV. This was the most challenging task for us as it took 17+ hours to train a single machine learning algorithm due to a very large dataset and numerous combinations of the algorithm’s parameters.

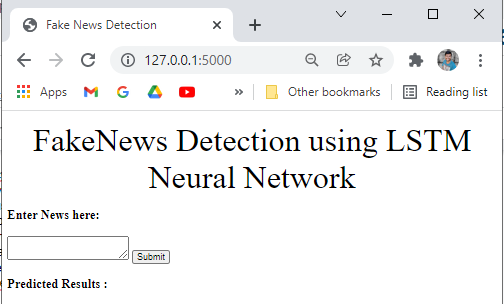
We found out that LSTM is the best model for our dataset, and we decided to use LSTM as our final ML algorithm. To make sure that our model is trained on equal splits of data, we have cross-validated a training dataset and ensured minimum bias and variance.

**Figure 8: Post Hyperparameter Tuning Model Evaluation Metrics.**

1. Model Evaluation

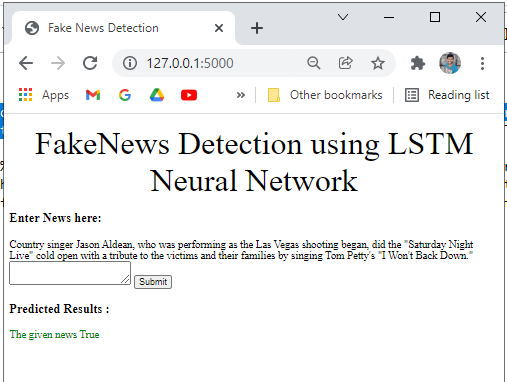
Model evaluation helps us to identify the performance of the machine learning model. We have various evaluation metrics for classification problems. Our model’s goal is to predict and classify whether a particular text is True or False; we deduce that accuracy would be the suitable evaluation metric for our problem.

1. Results

As per our approach, we have successfully exported the machine learning model into the pickle file used by Flask to communicate and classify whether the given news is True or False. Once the classification is done, Flask will receive a response from the ML model, and the response will be sent to the UI – visible to the end-user.

**Figure 9: Flask Front end - Home Page.**

As shown above, the user has to input the text into the textbox provided. This can be copied pasted from any portal as the web application can evaluate the textual data.



**Figure 10: Flask Front end - Result.**

# **References**

1. D. of J. Government of Canada, “Section 11(d) – Presumption of innocence,” Charterpedia, 25-Jan-2021. [Online].
2. [Webb et al.2016] Helena Webb, Marina Jirotka, Bernd Carsten Stahl, William Housley, Adam Edwards, Matthew Williams, Rob Procter, Omer Rana, and Pete Burnap. 2016. Digital wildfires: hyper-connectivity, havoc, and a global ethos to govern social media. ACM SIGCAS Computers and Society, 45(3):193–201.
3. [Wingfiled et al.2016] Nick Wingfiled, Mike Isaac, and Benner Kate. 2016. Google and Facebook aim fake news sites, November. [Online; posted Nov. 14, 2016].
4. Wang, W. Y. (2017).” liar, liar pants on fire”: A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648.
5. Zhang, J., Cui, L., Fu, Y., & Gouza, F. B. (2018). Fake news detection with the deep diffusive network model. arXiv preprint arXiv:1805.08751.
6. H. Gupta, M. S. Jamal, S. Madisetty and M. S. Desarkar, “A framework for real-time spam detection in Twitter,” 2018 10th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, 2018, pp. 380-383
7. C. Buntain and J. Golbeck, “Automatically Identifying Fake News in Popular Twitter Threads,” 2017 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, 2017, pp. 208-215.
8. Chen, Y., Conroy, N. J., & Rubin, V. L. (2015, November). Misleading online content: Recognizing clickbait as false news. In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection (pp. 15-19). ACM.
9. M. Granik and V. Mesyura, “Fake news detection using naive Bayes classifier,” 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903.
10. Helmstetter, S., & Paulheim, H. (2018, August). Weakly supervised learning for fake news detection on Twitter. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE.
11. M. L. Della Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro, and L. de Alfaro, “Automatic Online Fake News Detection Combining Content and Social Signals,” 2018 22nd Conference of Open Innovations Association (FRUCT), Jyvaskyla, 2018, pp. 272-279.
12. Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots. arXiv preprint arXiv:1707.07592, 96-104.
13. S. B. Parikh and P. K. Atrey, “Media-Rich Fake News Detection: A Survey,” 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, 2018, pp. 436-441
14. Schonfeld, E.: OpenGenus IQ: Learn Computer Science - Porter Stemmer algorithm. iq.opengenus.org Weblog Article (2009), https://iq.opengenus.org/porter-stemmer/.