# Fake News Detection Using Machine Learning Algorithms

Zahra Khaled Mahmoud-202000024

Omar Tarek Youssef-202000078

begad Soliman-202001947

Karen Ehab Lotfy-202002101

Nour- 202000383

# Problem Statement:

The introduction of the World Wide Web and the quick popularity of social media platforms (such as Facebook and Twitter) cleared the door for unprecedented knowledge distribution in human history. Aside from other applications, news organizations benefited from the extensive usage of social media platforms by offering consumers with up-to-date information in near real time. Newspapers, tabloids, and magazines gave way to digital types of media such as online news sites, blogging, social feeds, and other online media forms.

While the advantages of social media, the quality of content on social media is lower than that of conventional news organizations. However, since it is cheap to provide news online and significantly speedier and simpler to distribute via social media, enormous amounts of faux news, i.e., news pieces with purposefully misleading material, are generated online for a variety of reasons, including economic and political benefit. It is predicted that over 1 million tweets are linked to the false news "Pizzagate." "By the conclusion of the presidential election. Given the ubiquity of this new phenomena, the term "fake news" has been coined "In 2016, the Macquarie Dictionary designated it the word of the year.

Propagandists often manage fake news to communicate political ideas or influence. For example, some reports indicate that Russia has developed fake accounts and social bots to propagate misleading articles. Third, false news alters how people understand and respond to actual news. For example, some fake news was simply intended to instill distrust and confusion in people, making it difficult for them to distinguish what is authentic from what is not. To help limit the detrimental consequences of false news. It is critical that we develop tools for automatically detecting bogus news on social media.

This paper aims to predict if news is fake or not and comparing between traditional machine learning algorithms and deep learning algorithms. The study's purpose is to look at how these specific algorithms function for these specific problems and to verify the idea of employing AI for false news identification.

# Related work:

Himank Gupta et al. presented a framework built on several machine learning approaches in their study that addresses a variety of issues such as accuracy deficit, time lag (BotMaker), and increased processing time to accommodate thousands of tweets in one second. To begin, they gathered 400,000 tweets from the HSpam14 dataset. They next differentiate between 150,000 spam tweets and 250,000 non-spam tweets. They also extracted several lightweight features, as well as the Top-30 terms from the Bag-of-Words model that provide the most information gain. They achieved an accuracy of 91.65% and outperformed the old solution by almost 18%.

Granik, M.,& Mesyura,V. (2017) used a naïve Bayes classifier to detect bogus news. The authors describe a basic technique to false news identification using a naïve Bayesian classifier at the IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 900-903. This method is being evaluated using data taken from Facebook news posts. They claim to be able to reach a 74% accuracy. This model's performance is high, but not the greatest, since several other papers have reached a higher rate using other classifiers.

Aphiwongsophon et al. collected data from Twitter and analyzed them using advanced classifiers such as SVM and Neural Network models. Their testing revealed an accuracy rate of 99.90% in recognizing bogus news in specific areas.

Anshika Choudhary et al. 2020 investigated linguistic elements to determine if news is false or true. In their investigation, they investigated four language features: syntax-based, sentiment-based, grammatical, and readability-based evidence. They achieved an accuracy of 72% using classic machine learning-based ensemble approaches, but a progressive neural network-based approach improves the prior one and achieves an accuracy of 86%.

# - Methodology:

In our project, we used Python and its libraries like pandas, numpy, and sklearn. Python includes a large number of libraries and features that may be utilised in Machine Learning. Our project is centered on running multiple tests using various algorithms including MultinomialNB, Passive Aggressive Classifier, Sentiment Analysis, Losng Short Memory LSTM, logistic regression, and Decision Tree. We ran each model on the dataset and then measured its accuracy which is a measure of the overall correctness of the classifier, and it is calculated as the number of correct predictions divided by the total number of predictions made by the classifier , precision which is a measure of the accuracy of positive predictions, and it is calculated as the number of true positive predictions divided by the total number of positive predictions made by the classifier, a classifier with high precision has a low false positive rate, meaning that it is less likely to label a negative example as positive and the recall which is a measure of the completeness of positive predictions, and it is calculated as the number of true positive predictions divided by the total number of positive examples in the data, a classifier with high recall has a low false negative rate, meaning that it is less likely to label a positive example as negative.

# Implementation:

## Dataset:

The dataset contains two types of articles fake and real News. This dataset was collected from realworld sources; the truthful articles were obtained by crawling articles from Reuters.com (News

website). As for the fake news articles, they were collected from different sources. The fake news

articles were collected from unreliable websites that were flagged by Politifact (a fact-checking

organization in the USA) and Wikipedia. The dataset contains different types of articles on different

topics, however, the majority of articles focus on political and World news topics.

The dataset consists of two CSV files. The first file named “True.csv” contains more than 12,600

articles from reuter.com. The second file named “Fake.csv” contains more than 12,600 articles from

different fake news outlet resources. Each article contains the following information: article title, text,

type and the date the article was published on. To match the fake news data collected for kaggle.com,

we focused mostly on collecting articles from 2016 to 2017. The data collected were cleaned and

processed, however, the punctuations and mistakes that existed in the fake news were kept in the text.

Table

Description automatically generatedThe following table gives a breakdown of the categories and number of articles per category.

Figure

## Pre-processing Data:

### Data Cleaning:

1. Remove Stopwords:

Stopwords are frequent words that may be found in any writing. We eliminate them because they don't offer us much regarding our data. For example, silver or lead is ok with me-> silver, lead, fine.

1. Tokenization:

Tokenizing text divides it into parts like sentences or words. It provides formerly unstructured text structure. Plata o Plomo, for example, becomes 'Plata','o','Plomo'.

1. Remove punctuation:

Punctuation can help us grasp a phrase by providing grammatical context. However, because our vectorizer measures the amount of words rather than the context, it adds no value, therefore we eliminate all special characters. For example: How are you? ->How are you

### Vectorizing Data:

1. Vectorizing Data: N-Grams:

N-grams are simply all possible combinations of neighbouring words or letters of length n in our given text. Unigrams are ngrams with n=1. Likewise, bigrams (n=2), trigrams (n=3), and so on can be employed. Unigrams often carry less information than bigrams and trigrams. The core idea behind n-grams is that they catch the letter or word that is most likely to come just after supplied word. The more background you have to work with, the longer the n-gram (greater n).

1. TF-IDF:

It computes the "relative frequency" of a word appearing in a document in comparison to its frequency in all documents TF-IDF weight. TF refers to Term Frequency: It estimates how often a phrase appears in a document and shows the relative relevance of a term in the document and overall corpus. Because the size of each document changes, a phrase may feature more prominently in a long document than in a short one. As a result, the length of the text frequently often divides Term frequency.

IDF stands for Inverse Document Frequency: A term is useless if it appears in every document. Some phrases, such as "a," "an," "the," "on," "of," and so on, exist often in a document yet have little significance. IDF reduces the relevance of these phrases while increasing the importance of uncommon ones. The greater the value of IDF, the more distinctive the term.

## Algorithms used for Classification:

To anticipate the text's class, many classifiers were studied. We especially investigated four alternative machine-learning techniques-MultinomialNB, Passive Aggressive Classifier, Sentiment Analysis, Losing Short Memory LSTM, logistic regression, and Decision Tree.

## Implementation Steps:

Step 1: importing libraries, reding the file.

Step 2: Feature Extraction; we have extracted features from the already pre-processed dataset(text,label).

Step 3: Create Dataset1 for true and dataset2 for false news. Then merge the 2 datasets and remove null values.

Step 4: check if data is balanced to avoid overfitting.

Step 5: clean data.

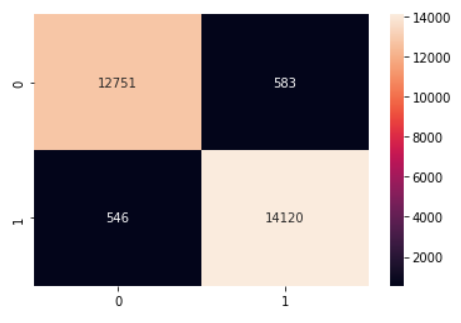
Step 6: vectorizing data.

Step 7: Here, we have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers.

Step 8: Once fitting the model, we get each model score, accuracy, recall and precision and confusion matrix.

# Results:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| model | MultinomialNB | Passive Aggressive Classifier | Sentiment Analysis | Losing Short Memory LSTM | logistic regression | Decision Tree |
| Accuracy score | 0.95 | 0.99 | 0.99 | 0.92 | 0.99 | 0.99 |

Confusion Matrix:

Chart, treemap chart

Description automatically generatedFigure : MultinominalNB

Figure : Passive Aggressive Classifier

A picture containing chart

Description automatically generated

Figure :LSTM