A Machine Learning Approach To Detect Covid-19 Fake Tweets

Xhoel Bano

Computer Engineering Department

Epoka University, Tirana, Albania

ABSTRACT

False information about COVID-19 has been spreading even quicker than the truth during the Covid-19 epidemic, a time when accurate information is essential for the public's health and safety. The scientific community has made significant efforts to address this issue because of the detrimental effects these news stories have on several fields. I want to contribute to the fight against the Covid-19 infodemic in this research by developing a probabilistic machine learning model for detecting fake information (tweets) that circulates on social media. This is why two datasets of tweets from Twitter were combined and compiled from numerous social media accounts taken into account. In this research, a tweet pre-processing pipeline designed for this particular categorization is shown. BOW and TF-IDF representations are used to extract numerical feature vectors from categorical input. K fold cross validation is used to train and assess the Multinomial Naïve Bayes classifier. The final outcomes are then predicted using an additional testing dataset. On the test set, the model achieves a 78.7 % accuracy rate and an F1-score of 78.4 %. Given the relative simplicity of the model, this is a respectable outcome in compared to the baseline techniques employed for this dataset.

Keywords: *covid-19, probabilistic model, machine learning, fake tweet detection, Naïve Bayes*

1 INTRODUCTION

The WHO classified the coronavirus (COVID-19) outbreak as a worldwide pandemic in March 2020. A COVID-19 infodemic is also happening right now, in which an excessive amount of information is being produced and disseminated throughout the whole planet. Misinformation about COVID-19 was common in the early stages of the pandemic as a result of ignorance about the virus and contentious discussions about the best ways to combat it. Regarding the cause, scope, prevention, diagnosis, and treatment of the disease, it contained erroneous or deceptive material and conspiracy theories [1]. Misinformation promotes untested therapies, generates confusion, and drives individuals to reject public health measures like vaccinations and masks [2].

We cannot avoid using social media sites like Facebook, Instagram, TikTok and Twitter in our everyday lives. These social media sites are useful tools for sharing information, including news, images, and other kinds of media. These platforms have advantages like ease, but they are also frequently used to spread dangerous material or information. This false information could lead consumers astray and potentially have harmful effects on the culture, economy, and healthcare of a society. It's challenging to stop the spread of this much false information. Therefore, the spread of false information about the COVID-19 pandemic, its treatment, and vaccine may provide serious difficulties for the frontline workers in each nation. Building a powerful machine learning (ML) misinformation detection model is therefore crucial for spotting the false information about COVID-19.

Researchers have previously attempted to detect fake news by text categorization using various machine learning-based NLP (natural language processing) methods. However, when applied to a particular domain, like the new corona virus, there are a number of problems that occur that make this identification a difficult process, such as the dearth of suitable annotated datasets [4], the constantly shifting body of knowledge regarding this virus, etc. A manually annotated dataset of 10,700 false and true social media news stories connected to COVID-19 has been detailed and made available by Patwa et al. in [4] to aid in the identification of fake news regarding this outbreak. Additionally, they benchmarked this dataset using four ML baselines: Gradient Boost, Decision Tree, Logistic Regression, and SVM. As part of a collaborative endeavour at the CONSTRAINT-2021 workshop, the dataset is made public. On the basis of this dataset, several analytical techniques are applied to investigate the categorization of news stories on the corona virus. In his study, Felber [5] used numerous language variables, including n-grams, readability, emotional tone, and punctuation, to analyse the performance of certain conventional ML models.

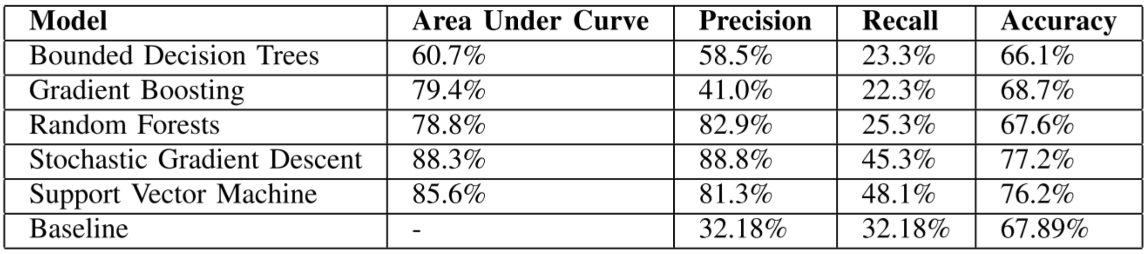
In this research proposed, I aim to develop a classification model for COVID-19 tweets, using the simple but effective Multinomial Naïve Bayes Algorithm and also combining three datasets. The first dataset “Coronavirus tweets NLP - Text Classification” is extracted from Kaggle [9], the second dataset Constraint@AAAI2021 workshop [4], and the third one “CoAID” [10]. My goal is to build a ML model based on Multinomial Naïve Bayes Algorithm and predict fake COVID-19 tweets from the datasets combined above. Also I intend that the performance of this traditional ML model to be compared with the more recent or even future, more sophisticated pre-trained language converters as well as the baseline techniques (SVM, DT, Logistic Regression, and Gradient Boost).

2 Literature Review

Various NLP based researches are conducted for text classification using different machine learning techniques. The primary goal of research of Ibrahim and Sayed [6] is to identify false news and disinformation about COVID-19 on the Internet. They have been testing with numerous methodologies and models in order to determine the best performing model. They employed a labeled data set including bogus news that can be recognized using cutting-edge natural language processing techniques and powerful deep learning algorithms. To determine the best feasible combination for the dataset, researchers compare the accuracy of simple approaches to contemporary and advanced techniques in the deep learning family. They give a labeled COVID-19 news article dataset with a classification model that predicts whether the news is real or fake in this study. After training, they have observed the Precision, Recall, F1 score and Test accuracy for each combination. It is observed that Bi-directional LSTM with Word2Vec as word embedding has gained a satisfactory test accuracy of 99.3% [6]. The hybrid LSTM+CNN model is the second best performing model which also has Word2Vec as word embedding method.

Meanwhile, in this paper, J.Khan, S.Afroz and K.Tawkat [7] conducted a benchmark study to evaluate the performance of several relevant machine learning algorithms on three separate datasets, of which we gathered the largest and most diverse. They investigated a variety of advanced pre-trained language models for false news detection, in addition to standard and deep learning models, and compared their performances in several aspects for the first time, to the best of our knowledge. They discovered that BERT and comparable pre-trained models perform well for detecting bogus news, especially with small datasets. As a result, these models are a much better choice for languages with less electronic content, i.e., training data. They also conducted many analyses depending on the models' performance, the topic of the article, the length of the article, and the lessons learnt from them. They anticipate that this benchmark study will assist the research community and news sites/blogs in selecting the best fake news detection algorithm [7]. On three separate datasets, they evaluated a wide range of machine learning algorithms, including both classical (e.g., SVM, LR, Decision Tree, Naïve Bayes, k-NN) and deep learning (e.g., CNN, LSTM, Bi-LSTM, C-LSTM, HAN, Conv-HAN) models. They created a new merged dataset including 80k news stories from multiple sources on a wide range of themes (e.g., politics, economy, investigation, health-care, sports, and entertainment). In their comparison research, they also looked at a number of pre-trained models, such as BERT, RoBERTa, Distil-BERT, ELECTRA, and ELMo. They discover that deep learning models outperform standard machine learning models in general. Among the traditional learning models, Naïve Bayes gets 93 % accuracy on combined corpus. Bi-LSTM and C-LSTM show remarkable potential among deep learning models, with 95 % accuracy on combined corpus [7].

A research is conducted by C. Liemeng and L. Dongwon [8] in order to create a proper dataset for text classification task related to covid-19 tweets and newses. They offered CoAID (Covid-19 Healthcare Misinformation Dataset) containing different COVID-19 healthcare misinformation, including fake news on websites and social platforms, as well as users' social interaction regarding such news, to assist academics tackle COVID-19 health misinformation. CoAID comprises 4,251 news items, 296,000 associated user engagements, 926 COVID-19-related social platform postings, and ground truth labels. The dataset may be found at https://github.com/cuilimeng/CoAID.



Model	Area Under Curve	Precision	Recall	Accuracy
Bounded Decision Trees	60.7%	58.5%	23.3%	66.1%
Gradient Boosting	79.4%	41.0%	22.3%	68.7%
Random Forests	78.8%	82.9%	25.3%	67.6%
Stochastic Gradient Descent	88.3%	88.8%	45.3%	77.2%
Support Vector Machine	85.6%	81.3%	48.1%	76.2%
Baseline	-	32.18%	32.18%	67.89%

Table 1. *Average model performances with only TF-IDF bi-gram features at 0.7 score threshold for categorization [8]*

On the other hand, in their study [4], Patwa et al. discuss and make available a manually annotated dataset of more than 10,000 false and real social media news articles connected to COVID19. This dataset relates to the particular Covid-19 infodemic. By doing this, they hope to assist researchers in addressing the COVID-19 Infodemia issue. Since true news makes up 52.34 % of the samples and fraudulent news makes up 47.66 %, the dataset is class-wise balanced. By keeping the class-wise distribution, the dataset is divided into train (60 %), validation (20 %), and test (20 %). They discovered through exploratory data analysis (EDA) that there is a large overlap between key terms in false and authentic news. They used the term frequency-inverse document frequency (TF-IDF) statistical approach as a feature extraction technique. Then, they use machine learning methods to benchmark the created dataset, projecting them as potential baselines: Support Vector Machine (SVM) with a linear kernel, Logistic Regression (LR), Gradient Boost and Decision Tree (GDBT). They observed that the SVM-based classifier among the ML models yields a higher accuracy and F1-score of 93.46 %.

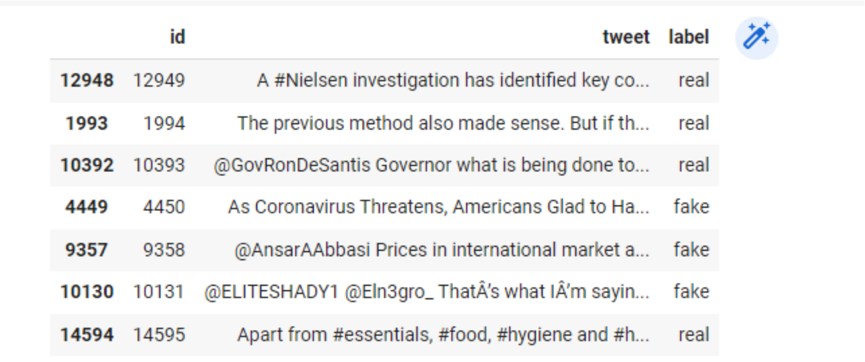
In his paper, Felber [5] applied traditional machine learning algorithms, including SVM, Random Forest, Logistic Regression, Naïve Bayes, and Multilayer Perceptron, along with a number of linguistic features, including n-grams, readability, emotional tone, and punctuation, to tackle this classification problem. They sought to outperform the Patwa et al. [4] baseline technique by using these additional linguistic factors. In fact, by using linguistic variables, they were able to produce the best F1-score on test data for their SVM model, which was 95.19 %.

3 Materials

3.1 Dataset

To develop a classification model for COVID-19 news (tweets), using the simple but effective Multinomial Naïve Bayes Algorithm I have combined three datasets. The first dataset “Coronavirus tweets NLP - Text Classification” is extracted from Kaggle [9], the second dataset extracted from Constraint@AAAI2021 workshop [4], and the third one “CoAID” [10].

This dataset is made up of tweets from extracted from Twitter and represent actual or fraudulent news on the COVID-19 pandemic. The dataset includes information about each tweet's id, which is not essential to our classification, as well as two categorical values: the tweet's text and the class that it belongs to. The two class labels for this binary classification problem are fake and real, as was previously indicated. 35,024 tweet samples make up the dataset. There are no missing values to contend with because it is personally curated from the respective writers. A portion of this dataset is displayed in Figure 1.



	id	tweet	label
12948	12949	A #Nielsen investigation has identified key co...	real
1993	1994	The previous method also made sense. But if th...	real
10392	10393	@GovRonDeSantis Governor what is being done to...	real
4449	4450	As Coronavirus Threatens, Americans Glad to Ha...	fake
9357	9358	@AnsarAAbbasi Prices in international market a...	fake
10130	10131	@ELITESHADY1 @Eln3gro_ ThatÂ’s what IÂ’m sayin...	fake
14594	14595	Apart from #essentials, #food, #hygiene and #h...	real

Figure 1. *Dataset sample*

By keeping the class-wise distribution, the dataset was initially divided into train (60 %), validation (20 %), and test (20 %). I have given a class-wise distribution split across the training shown in [Figure 2], test shown in [Figure 3], and validation shown in [Figure 4].

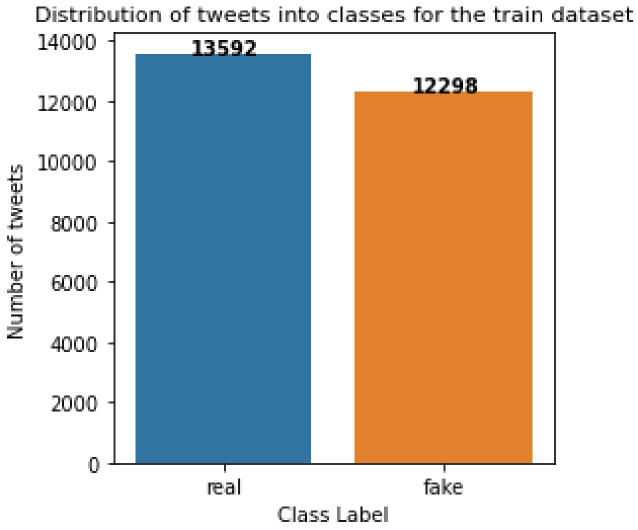


Figure 2. Distribution of tweets into classes for the train dataset

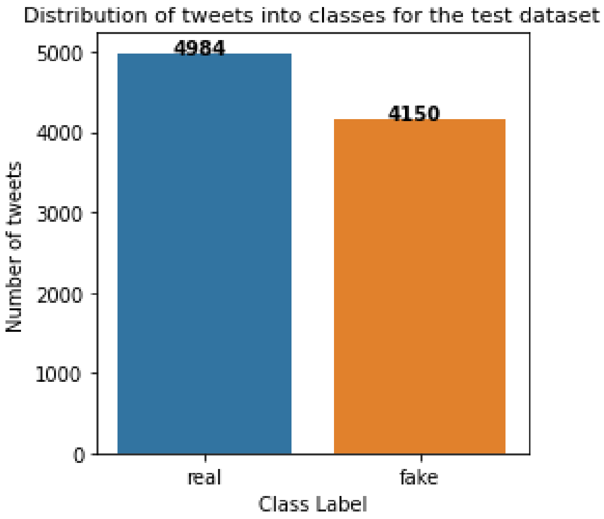


Figure 3. Distribution of tweets into classes for the test dataset

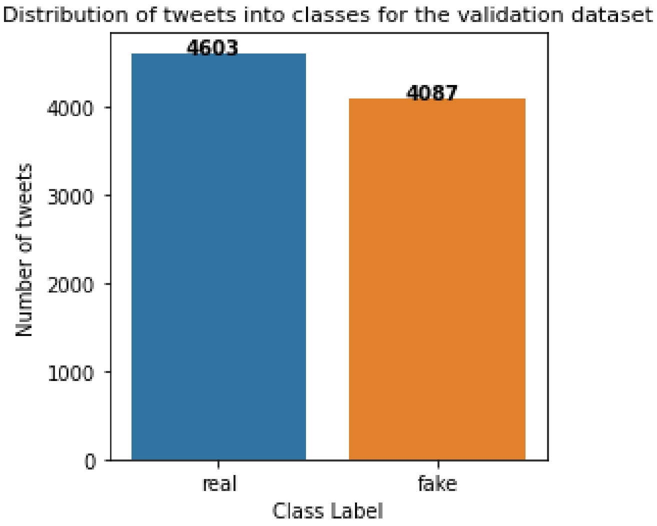


Figure 4. Distribution of tweets into classes for the validation dataset

3.1.1 Pre-processing steps

Text pre-processing is regarded as a critical stage in natural language processing activities. The goal of text pre-processing is to obtain a 'clean' version of the text suited for our job at hand by eliminating any unneeded or unclear patterns within the text. This section explains the fundamentals. This paper is based on a pre-processing pipeline [Figure 5]. This pipeline was built for a single tweet, then applied to all of the samples in our dataset:

1. **Lowercasing**: Lowercasing is the first stage in tweet pre-processing. Each word is lowercased such that the same term is treated the same way whether it is uppercased or lowercased.

2. **Remove URL links**: URL references in news material on social media are frequently related with redirects to other sources. These mentions add nothing to our categorization, but they may contribute to an inaccurate classification if they include contradictory terms.
3. **User Mentions Removal:** Because all user mentions in tweets contribute no value to our categorization, they may be safely eliminated.

4. **Expansion of contractions:** Contractions are abbreviated forms of words. They are both spoken and written in the English language. They are frequently produced by eliminating one of the word's vowels. Converting contractions to their expanded form would aid in text standardization.

5. **Tokenization:** Tokenization is the process of dividing a text into distinct tokens (words).

6. **Punctuation removal:** This step is performed after the URL and user mentions have been normalized since the links would otherwise be divided into sub-tokens and our regular expressions would fail to match the links effectively.

7. **Removal of non-alphabetic words:** Any word containing non-alphabetic letters is eliminated. An exception is allowed for the term 'covid-19.' This term is retained despite the fact that it contains numbers since it communicates critical meaning for our categorization and should not be eliminated.

8. **Stop-word Filtering:** A stop word is a commonly used term in natural language such as a, an, the, do, and so on. By removing stop words, we would be left with fewer and only relevant tokens, reducing the dataset size, increasing performance, and enhancing overall performance.

9. **Lemmatization:** This is the process of determining a word's lemma (the basic form depending on context). We aggregate the many forms of a word together so that they may be analysed as a single characteristic.

10. **Short words removal:** All leftover words with fewer than three characters are deleted in this phase. These might be meaningless words that resulted from the omission of punctuation.

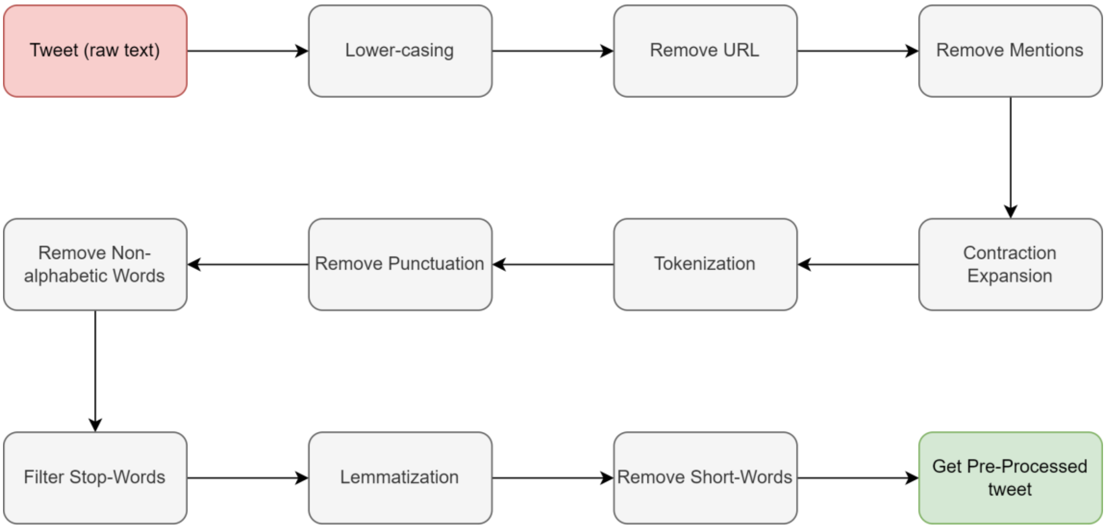


Figure 5. *Tweet pre-processing pipeline (steps)*

3.1.2 Feature extraction

To perform machine learning on text data, we require feature extraction algorithms to transform each tweet into a numerical feature vector that the model can interpret. The Naïve Bayes classifiers' probabilistic model is based on Bayes' theorem, and the word Naïve stems from the assumption that the characteristics in a dataset are mutually independent. To extract mutually exclusive features from text input and compute their probabilities for use with the Naïve Bayes classifier, I have approached from two perspectives:

3.1.2.1 Term Frequency times Inverse Document Frequency (TF-IDF)

The Bag Of Words (BOW) model is a simplified representation that is used to facilitate natural language processing and information retrieval (IR). The BOW model represents a text as an unordered collection of its words, ignoring syntax and even word order [11]. Each individual word from the whole training dataset is allocated a weight based on its frequency of occurrence in each sample. Each occurrence is utilized to train the classifier as a feature. In my scenario, each individual word in any tweet is allocated a constant integer Id. The number of occurrences of the word 'w' in that tweet is then computed for each tweet and saved in position X[i][w], where X[i] is the feature vector representation of tweet "i". [12]

3.1.2.2 Term Frequency times Inverse Document Frequency (TF-IDF)

The TF-IDF is a numerical statistic that indicates how essential a certain word is to a specific text in a collection or corpus [13]. It works by comparing the relative frequency of terms in a single text to the inverse fraction of that word over the whole corpus of documents [14]. Our BOW model's TF-IDF representation is computed in two steps:

1. Term Frequency: We divide the number of occurrences of each term in a tweet by the total number of words in that tweet to calculate Term Frequency. In this manner, we regard the frequencies rather than the occurrences as characteristics. This normalization is required in the case of long tweets, which may have larger count values than shorter tweets despite discussing the same themes.

$$TF = \frac{Number\ of\ times\ word\ \boldsymbol{w}\ appeard\ on\ tweet\ \boldsymbol{i}}{Total\ number\ of\ words\ in\ tweet\ \boldsymbol{i}} \qquad (1)$$

2. Inverse Document Frequency: A further refinement used on top of TF is to weight down common words and scale up unusual ones. According to information theory, the more common a term is in tweets, the less informative it is in compared to those that occur in a tiny percentage of the corpus. The following formula yields the IDF representation of tweets:

$$IDF = \log(\frac{Number\ of\ tweets}{Number\ of\ tweets\ with\ word\ \boldsymbol{w}\ in\ it}) \qquad (2)$$

3.2 Software Components

This project is written in Python and built with Google Colaboratory. Python libraries such as pandas [15] for dataset loading and data analysis, seaborn and matplotlib [16] for data visualization, nltk [17] for text pre-processing, and sklearn [12] for feature extraction, Naïve Bayes implementation, and classification metrics are used in the implementation.

4 Methods

4.1 Naïve Bayes Algorithm

The Multinomial Naïve Bayes algorithm is the supervised classifier utilized in this study for false tweet (text) classification. Naïve Bayes is a classifier family based on the well-known Bayes' probability theorem. The Bayes' Theorem calculates the likelihood of an event occurring given the chance of another event occurring [18]. The following formula represents it:

$$P(A|B) = \frac{P(A|B)\ P(A)}{P(B)} \qquad (3)$$

P(A|B) represents the posterior probability that must be determined, P(B|A) represents the conditional probability or likelihood, P(A) represents the prior probability, and P(B) represents the evidence [19]. Naïve Bayes is a probabilistic classifier, which means that for a document d, the classifier gives the class with the highest posterior probability given the document. This is expressed mathematically as:

$$\hat{c} = argmax\ P(c|d)\ , \quad c \in C \qquad (4)$$

4.1.1 Multinomial Naïve Bayes Classifier for tweet classification

Assume we want to determine if a particular tweet is real or fake. We must compute the odds that the class is fake given the words in the tweet, as well as the probabilities that the class is real given the words in the tweet. These two probabilities must be compared in order for the model to predict the class of a given tweet.

In this section is described how the Multinomial Naïve Bayes method is used to my particular case. To use Multinomial Naïve Bayes for text classification, we require a collection of features and the probabilities for each of their values. Multinomial Naïve Bayes takes into account a feature vector in which the supplied term reflects the number of times a word appears in a document (tweet in our example), or, more often, the frequency [20]. Using the feature selection process described in section 3, each tweet is represented as a feature vector using the BOW model and improved with the Term Frequency - Inverse Document Frequency representation. We will utilize the data frequencies to learn the likelihood of feature fi given class c: P(fi | c).

It is assumed that a feature fi is just the existence of a word w_i in the tweet's bag of words. So, we are interested in the finding probability that the news is fake when the tweet contains words like w_1 w_2 … w_n. This can be represented as P(fake | w_1, w_2, … w_n). To break down the problem further, a naïve assumption is made: the words (features) w_1, w_2, … w_n are considered to be independent of each other. Thus, based on Bayes theorem, the probability that the news is fake can be calculated as:

$$\text{P(fake} \mid w_1, w_2, \dots\ w_n) = \frac{\text{P(fake)}\cdot\ [\ \text{P}(w_1|\text{fake})\ \cdot \text{P}(w_2|\text{fake})\ \cdot\dots\cdot\ \text{P}(w_n\ |\text{fake})\ ]}{\text{P}(w_1,w_2,\dots\ w_n)} \qquad (5)$$

In order to make the calculation in equation (5), we decompose it. First find and calculate the prior probability P(fake) with the formula below:

$$\text{P(fake)} = \frac{N_{fake\ tweets}}{N_{tweets}} \qquad (6)$$

Then use the formula below (7) to find the likelihood for a feature P(f_i | fake):

$$\text{P}(\text{w}_i|\ \text{fake}) = \frac{\text{Count of w}_i\text{ in fake tweet}}{\text{Total number of features in fake tweet}} \qquad (7)$$

Do the same steps for the real class using the equations (8), (9) and (10). Since, the posterior probability for both classes will be compared, the evidence P(w_1, w_2,... w_n) for both classes may be discarded. The best projected class for a particular tweet is determined by the class with the highest posterior probability.

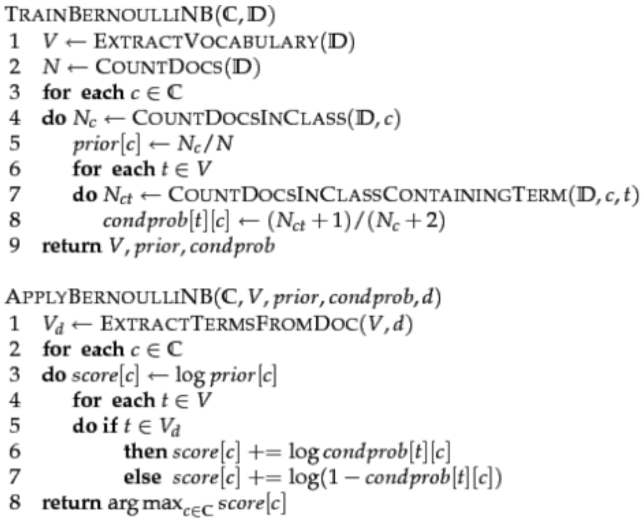
$$\text{P(real} \mid w_1, w_2, \dots\ w_n) = \frac{\text{P(real)}\cdot\ [\ \text{P}(w_1|\text{real})\ \cdot \text{P}(w_2|\text{real})\ \cdot\dots\cdot\ \text{P}(w_n\ |\text{real})\ ]}{\text{P}(w_1,w_2,\dots\ w_n)} \qquad (8)$$

$$\text{P(real)} = \frac{N_{real\ tweets}}{N_{tweets}} \qquad (9)$$

$$\text{P}(\text{w}_i|\ \text{real}) = \frac{\text{Count of w}_i\text{ in real tweet}}{\text{Total number of features in real tweet}} \qquad (10)$$

4.1.2 Pseudocode

Figure X below represents the pseudocode for Multinomial Naïve Bayes Algorithm [21].



```
TRAINBERNOULLINB($\mathbb{C}, \mathbb{D}$)
$V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
$N \leftarrow$ COUNTDOCS($\mathbb{D}$)
for each $c \in \mathbb{C}$
do $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
    $prior[c] \leftarrow N_c / N$
    for each $t \in V$
    do $N_{ct} \leftarrow$ COUNTDOCSINCLASSCONTAININGTERM($\mathbb{D}, c, t$)
        $condprob[t][c] \leftarrow (N_{ct} + 1)/(N_c + 2)$
return $V, prior, condprob$

APPLYBERNOULLINB($\mathbb{C}, V, prior, condprob, d$)
$V_d \leftarrow$ EXTRACTTERMSFROMDOC($V, d$)
for each $c \in \mathbb{C}$
do $score[c] \leftarrow \log prior[c]$
    for each $t \in V$
    do if $t \in V_d$
          then $score[c]$ += $\log condprob[t][c]$
          else $score[c]$ += $\log(1 - condprob[t][c])$
return $\arg\max_{c \in \mathbb{C}} score[c]$
```

***Figure 6.** Pseudocode for Multinomial Naïve Bayes Algorithm [21]*

4.2 Validation Method

The k-fold cross validation procedure with k = 5 is utilized to evaluate this work. The combined dataset is randomly divided into k equal-sized folds (subsets). The model is then trained and tested k times. For each fold, we include the accuracy, recall, precision, and F1-score, as well as the average. The number of right guesses divided by the total number of forecasts is used to measure accuracy. Accuracy is defined as the ratio of correct positive predictions to total positive predictions. The ratio of correct positive predictions to the total number of positive labels is used to determine recall, and the F1-score is the weighted average of accuracy and recall.

5 Results

The results of running the preceding processes on the provided dataset are demonstrated below. The Multinomial Naïve Bayes model is trained using 25890 tweets (train and validation datasets) and tested with 5-Fold cross validation. Each tweet is input into the model as a 36266 dimension feature vector derived from the pre-processed tweets using BOW + TF-IDF. Table 2 presents the assessment criteria for each test fold as well as the overall mean.

Metric (mean)	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
Accuracy (%)	83	77	72	80	74	77.2
F1 score (%)	83	76	72	80	74	77
Precision (%)	85	77	72	81	74	77.8
Recall (%)	82	76	72	80	73	76.6

***Table 2.** 5-fold cross validation scores*

After training on the training data, the model is applied to the test set, which was never seen during training. On this test set, the performance is tested once more. This is done to ensure that the model has learnt more than simply the training data.

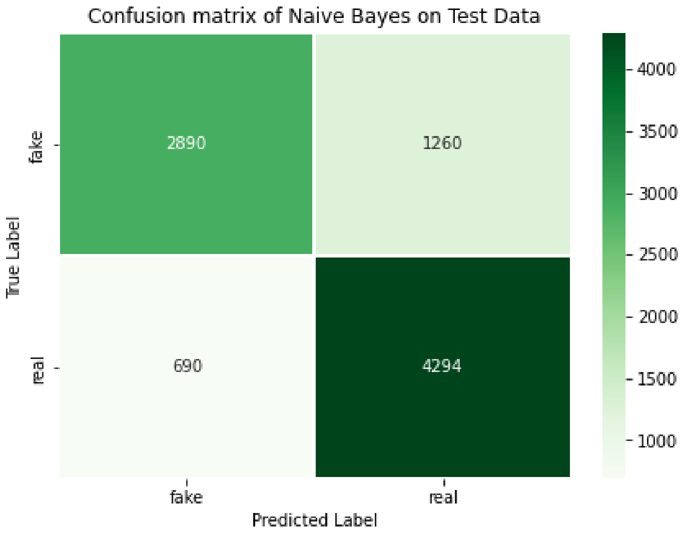
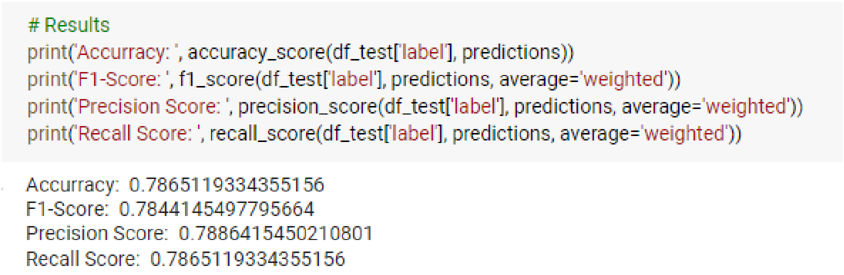


Figure 7. Confusion matrix

According to the confusion matrix, there are 2890 True Positives (tweets that are fake and are predicted as fake), 4294 True Negatives (tweets that are real and are predicted as real), 1260 False negatives (tweets that are fake but are predicted as real), and 690 False positives with respect to the fake class (fake = 1, real = 0). (tweets that are real but are predicted as fake). Accuracy, precision, recall, and F1-scores may be calculated using these findings. Figure 6 depicts the results from the program for both courses.

My model managed to get an accuracy of 78.7 %, a F1-Score of 78.4 %, Precision Score of 78.9 % and a Recall Score of 78.7 % [Figure X].



```
# Results
print('Accurracy: ', accuracy_score(df_test['label'], predictions))
print('F1-Score: ', f1_score(df_test['label'], predictions, average='weighted'))
print('Precision Score: ', precision_score(df_test['label'], predictions, average='weighted'))
print('Recall Score: ', recall_score(df_test['label'], predictions, average='weighted'))
```

```
Accurracy:  0.7865119334355156
F1-Score:  0.7844145497795664
Precision Score:  0.7886415450210801
Recall Score:  0.7865119334355156
```

Figure 8. Final results of the model

6 Conclusion

The purpose of this work was to aid in the fight against the Covid-19 infodemic by developing a machine learning model for the categorization of fake news propagating on social media. A dataset of 10700 false and real tweets is used in the analysis. The tweets are pre-processed and converted into a clean format using a natural language processing pipeline. Lowercasing, tokenization, URL normalization, username removal, contractions expansion, stop-word filtering, and short words removal are all part of this process. BOW and TF-IDF are employed as feature extraction algorithms to turn each tweet into a V-dimensional feature vector, where V is the vocabulary for the training set.

The ML method tested in this research is Multinomial Naïve Bayes, which is appropriate for classifying dissimilar features (word count for text classification). Considering that it is one of the simplest Machine Learning Models that just captures the probabilities of occurrence of words, this model produced a premising result of 78.7 % accuracy and an F1- score of 78.4 %. Running this implementation was simple because Naïve Bayes is a quick and accurate prediction algorithm with a cheap computational cost.

References

[1] Wikipedia. COVID-19 misinformation, 2021.

[2] Office of the Surgeon General et al. Confronting health misinformation: The US Surgeon General's advisory on building a healthy information environment [Internet]. 2021.

[3] Mohammed N. Alenezi & Zainab M. Alqenaei, 2021. "Machine Learning in Detecting COVID-19 Misinformation on Twitter," Future Internet, MDPI, vol. 13(10), pages 1-20, September.

[4] P. Patwa et al., "Fighting an Infodemic: COVID-19 Fake News Dataset," arXiv:2011.03327 [cs.CL], Nov 2020.

[5] T. Felber, "Constraint 2021: Machine Learning Models for COVID-19 Fake News Detection Shared Task," arXiv:2101.03717 [cs.CL], 2021.

[6] S. Ibrahim,S. Shoaib "COVID-19 Related Fake News Detection Model" 2021, bachelor thesis, doi: 10.3389/fpubh.2021.788074

[7] J. Y. Khan, M. T. I. Khondaker, S. Afroz, G. Uddin, και A. Iqbal, 'A benchmark study of machine learning models for online fake news detection', Machine Learning with Applications, τ. 4, σ. 100032, 2021.

[8] Cui, Limeng & Lee, Dongwon. (2020). CoAID: COVID-19 Healthcare Misinformation Dataset. pages 1-11, 2020, arXiv:2006.00885v3

[9] Aman Miglani, "Coronavirus tweets NLP - Text Classification" dataset, https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification?resource=download (lastly visited on 30 May 2022).

[10] L. Cui και D. Lee, 'CoAID: COVID-19 Healthcare Misinformation Dataset', arXiv [cs.SI]. 2020.

[11] G. K. Soumya and J. Shibily, "Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature," IOSR Journal of Computer Engineering (IOSR-JCE), vol. 16, no. I, pp. 34-38, 2014

[12] Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR 12, pp. 2825-2830, 2011.

[13] M. Apra and V. Santosh, "Analysis of TF-IDF Model and its Variant for Document Retrieval," in 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 2015

[14] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," Proceedings of the first instructional conference on machine learning, vol. 242 (1), pp. 29-48, 2003.

[15] McKinney, "Data structures for statistical computing in python," Proceedings of the 9th Python in Science Conference, vol. 445, 2010.

[16] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[17] S. Bird, L. Edward and K. Ewan, Natural Language Processing with Python, O'Reilly Media Inc., 2009.

[18] S. Raschka, “Naive Bayes and Text Classification I, Introduction and Theory,” arXiv:1410.5329v4 [cs.LG], 2014.

[19] J. Daniel &. J. H. Martin., “Naive Bayes and Sentiment Classification,” Speech and Language Processing, 2020.

[20] A. M. K. F. P. Holmes, “Multinomial Naive Bayes for Text Categorization Revisited,” Lecture Notes in Computer Science LNCS, vol. 3339.

[21] C. D. Manning, P. Raghvan and H. Schulze, Introduction to information retrieval, Cambridge: Cambridge University Press, 2008.

APPENDIX

Appendix A

Github repository (Dataset, Code, Research): https://github.com/xhoel-bano/A-Machine-Learning-Approach-To-Detect-Covid-19-Fake-Tweets