**Machine Learning Project Report   
on**

**Fake News Classifier System**

Submitted by: **Shubhranshi Kapoor**Roll no: **109164422**

Institute of Information Security  
National Tsing Hua University, Hsinchu, Taiwan

2021-2022

**Abstract**

Fake news detection has become tremendously important with the increase in the dependency on social media platforms for news sources. Platforms like Facebook and Twitter have millions of active users that receive and distribute articles without doing any fact check. The fake articles are solely created with malicious intentions to destroy the reputation of an individual or organization and for spreading hatred among communities. It can pose significant impact on our personal and social life and can also cause nationwide impact. Fake news detection techniques can help individuals and media houses to filter out such articles and stop their spread. Since the analysis of natural language text is a complicated process, developing efficient techniques is a challenging task. Fake news detection is one of the popular research topics in the machine learning community.

This work aims to solve this challenging task by applying various machine leaning models and natural language processing techniques. This study uses a dataset of labelled news articles from Kaggle consisting of both fake and true news articles. Natural language processing techniques like stop-word removal, stemming and n-gram approach with TF-IDF Vectorizer have been applied. The dataset is trained on different classifiers to investigate which of them works well for this specific task. This work explores the performance of some of the machine leaning models like Logistic Regression, Multinomial Naïve Bayes Classifier, Passive Aggressive Classifier and neural-network based learning model like LSTM (Long short-term memory) model. Performance metrics like precision, recall and F1-score are used to determine the performance of different machine learning models. For the fake news classification task on this Kaggle dataset, the Passive-Aggressive Classifier model gives the best performance. It achieves both accuracy and F1-score of 99%.

**Table Of Content**

[**Abstract** 2](#_Toc89304351)

[**Table Of Content** 3](#_Toc89304352)

[**Table Of Figures** 4](#_Toc89304353)

[**1** **Objective** 5](#_Toc89304354)

[**2** **Motivation** 6](#_Toc89304355)

[**3** **Related Work** 8](#_Toc89304356)

[**4** **Data Set** 10](#_Toc89304357)

[4.1 Data Source 10](#_Toc89304358)

[4.2 Data Exploration 10](#_Toc89304359)

[**5** **Data Preprocessing** 12](#_Toc89304360)

[**6** **Data Visualization** 13](#_Toc89304361)

[**7** **Feature Engineering** 14](#_Toc89304362)

[**8** **Machine Learning Models** 17](#_Toc89304363)

[8.1 Multinomial Naïve Bayes Algorithm:​ 18](#_Toc89304364)

[8.2 Logistic Regression Algorithm:​ 19](#_Toc89304365)

[8.3 Passive Aggressive Classifier Algorithm:​ 20](#_Toc89304366)

[8.4 LSTM (Long short-term memory) Algorithm:​ 22](#_Toc89304367)

[**9** **Results** 23](#_Toc89304368)

[**10** **Conclusion** 24](#_Toc89304369)

[**11** **References** 25](#_Toc89304370)

**Table Of Figures**

[Figure 1. First 5 records in the fake news articles dataset 11](#_Toc89299678)

[Figure 2. First 5 records in the true news articles dataset 11](#_Toc89299679)

[Figure 3. Pie-chart depicting distribution between fake and true news 13](#_Toc89299680)

[Figure 4. Bar-chart depicting subject wise news distribution 13](#_Toc89299681)

[Figure 5. First 10 features from the TF-IDF Vectorizer 15](#_Toc89299682)

[Figure 6. Overview of the processed workflow 17](#_Toc89299683)

[Figure 7. Confusion Matrix for Multinomial Naive Bayes Classifier 19](#_Toc89299684)

[Figure 8. Classification report for Multinomial Naive Bayes Classifier 19](#_Toc89299685)

[Figure 9. Confusion matrix for Logistic Regression Classifier 20](#_Toc89299686)

[Figure 10. Classification report for Logistic Regression Classifier 20](#_Toc89299687)

[Figure 11. Confusion matrix for Passive Aggressive Classifier 21](#_Toc89299688)

[Figure 12. Classification report for Passive Aggressive Classifier 21](#_Toc89299689)

[Figure 13. Model Summary for LSTM neural network model 22](#_Toc89299690)

1. **Objective**

The objective of this work is to build Fake News Classifier using a Machine Leaning model, which is trained on a dataset containing fake and real news files. The classifier will predict whether a given news article is fake or real. Some of the Natural Language Processing techniques are employed to deal with the textual news data.

1. **Motivation**

In recent times, Social media platforms have gained immense popularity. They have become an integral part of our daily lives. But, just as every coin has two sides, the content delivered through social media has pros as well as cons. Social media is easy to access, low cost and provides freedom to individuals in the sense of the right to speech. However, this comes with the cost of the spread of misinformation which often causes chaos and confusion. We faced the Covid-19 worldwide pandemic recently. In those times of uncertainty and fear, news and information, be it fake or true has played a significant role. Ever since the pandemic hit the world, there have been large volumes of misleading and false information about the effects of the virus, modes of its spread and the ways of curing it. Many social media platforms served to spread myths, which caused a state of panic and led people to resort to experimentation with home remedies. People, owing to a lack of knowledge, even resorted to making vaccines at home. The situation worsened when there was a spread of false news about some medicine, injection or medical equipment getting out-of-stock which resulted in people buying them in ample quantities even when they aren’t needed. This, in turn, caused the resources to get out-of-stock and deprive those in need. The same happened with food items and directly affected the less privileged sections of society. When dealing a national emergency, fake news creates more chaos and panic within citizens and increases problem for the authorities. Thus, it will not be an exaggeration to say that “infodemic” has taken far more lives than the pandemic itself.

Fake News plays a very important role in times of elections. It can cause wrong public opinions based on a piece of incorrect news or information, leading to majority votes being given to some candidate or party. These days, the News Media Houses have deviated away from the principle of conveying “true” news. They are just in search of any eye-catching news and start propagating that information without paying much heed to verification. Even before the detailed investigation is carried out, the verdict is ironically announced by the journalists themselves. This plays a significant role in forming a common opinion. One side is victimized while the other side is announced guilty by the public only. This is because our informers, be the journalists or the social media platforms, convince us to think in a certain direction. The sad part is that in this process, the facts get lost. A major reason for such an easy spread of fake news is that the content creators are generally free individuals, without the fear of any strict actions to be taken against them. Thus, there is no credibility of most of the content that one comes across on social media platforms.

Social media platforms serves as one of the major sources for fake news generation and propagation. Since these platforms are intentionally not news platforms but avenues for business for advertising, they offer little to no support for any fact check mechanism. Misinformation, displayed on News Programs not only wastes the time and energy of the viewer but also often hurts their sentiments. It propagates negative energy and leads to polarization and discrimination among people. It promotes the feeling of hatred and a sense of insecurity. Misinformation can often be held responsible for hideous acts like mob lynching, which has become quite a serious issue in recent years. Many platforms have started to use the concept of spreading misinformation as a means of earning. The concept of “click-baits” has evolved tremendously. There are many YouTube video links or links to some random website, which come with an eye-catching and often surprising title. As soon as the user clicks to open the video or the website, the content provided beats around the bush instead of giving any facts. When society has stooped so low as to increase the number of views of some video or website, without providing any authentic information, it becomes very essential to shift our focus on the detection of false news to prevent its propagation. For this, we need efficient Fake News Classification Algorithms that can be applied to as many platforms as possible.

1. **Related Work**

Fake news has emerged as one of the unfathomable problems in this current digital age. There have been various instances where people have harmed themselves and others under the influence of such news articles. These fake articles have a negative impact of an individuals’ well-being and also impact the harmony within the societies [1]. Especially when dealing with the COVID-19 pandemic, misinformation creates more panic among the citizens and authorities have an additional burden to combat with such fake articles [2]. There have been many instances across the world where people relied on untrue and unverified news sources for COVID-19 cure and caused more harm to their health than any benefit.

Fake news detection and classification is one of the popular machine learning tasks. Many researchers have tried various machine learning algorithms to address this challenging problem. Some scholars have developed automated methods to classify fake and true news stories from Twitter by using various features that are used by journalists for fact-checking [3]. Machine leaning algorithms like Naïve Bayes Classifier have also been used by some researchers for fake news detection to examine how well can these algorithms perform for this task. A simple fake news classifier based on Naïve Bayes can achieve an accuracy of 74% [4]. This suggest that machine learning models can be quite useful for handling this classification task. Deep Learning models like recurrent neural network, long short-term memory network, convolution neural networks for text have also given good results for the fake news classification tasks [5] [6]. Some authors have developed techniques for early detection of fake news. They use neural networks to generate comments that can help to evaluate the credibility of a news article [7].

Since news stories are written in natural language, using natural language processing techniques for text analysis can help in classification techniques. Some studies suggest that since fake stories are created for political and financial gains and to defame an entity, such posts mostly contain derogatory language and is just meant for click-bait. Extracting such linguistic features and writing styles can help to detect fake news. Also, capturing user profiles as one of the features can also help in fake news detection because a user that generates fake story once is more likely to produce more fake news. This can help in detecting and reducing fake news at the source level [8].

Some researchers have proposed n-gram feature approach to classify news articles as fake or true. Word based n-gram model can be used to generate features from the given text. These features are then given as input to various machine learning models to assess their performance [9]. Feature representation techniques like Term Frequency (TF) and Inverse Document Frequency (IDF), collectively denoted as TF-IDF can be used to generate a weighted vector representation of the news text. This feature vector can then be used as input to the machine learning classifier [10]. Various data pre-processing stages like removing stop-words, stemming and feature extraction helps to clean the data and improve the overall performance of machine learning models [11].

1. **Data Set**
   1. Data Source

In this project, Fake and Real News dataset from Kaggle is used for this Fake News Classification task [13]. The link to the dataset is given below:

<https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset>

Below files are given in this dataset:

* fake.csv​ – collection of fake news
* true.csv​ – collection of real news
  1. Data Exploration

The numpy and pandas libraries in python are used to load the data from csv files to python dataframe.

**fake.csv dataset –**

The fake.csv file contains 23481 fake news articles. The dataset is created for this file is ‘fake\_df’. It contains the following fields –

* **title** (object) : The title of the article.
* **text** (object) : The text of the article.
* **subject** (object) : The subject of the article.
* **date** (object) :The date at which the article was posted.

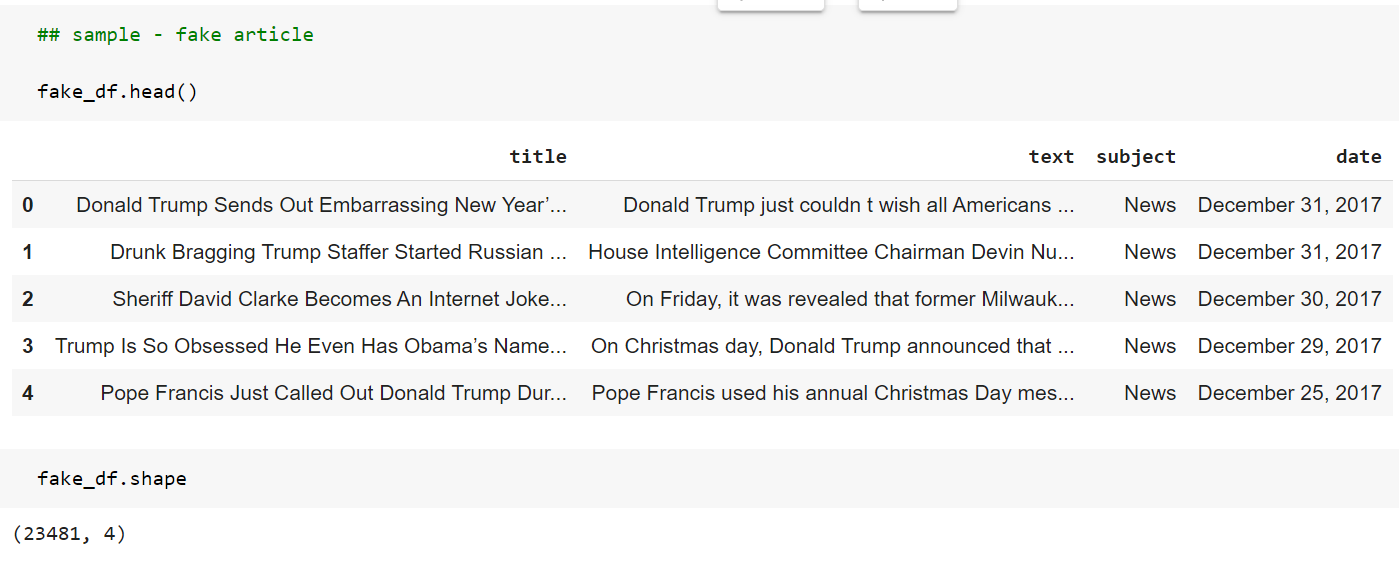


Figure 1. First 5 records in the fake news articles dataset

**true.csv dataset –**

The true.csv file contains 21417 true news articles. The dataset is created for this file is ‘true\_df’. It contains the following fields –

* **title** (object) : The title of the article.
* **text** (object) : The text of the article.
* **subject** (object) : The subject of the article.
* **date** (object) :The date at which the article was posted.

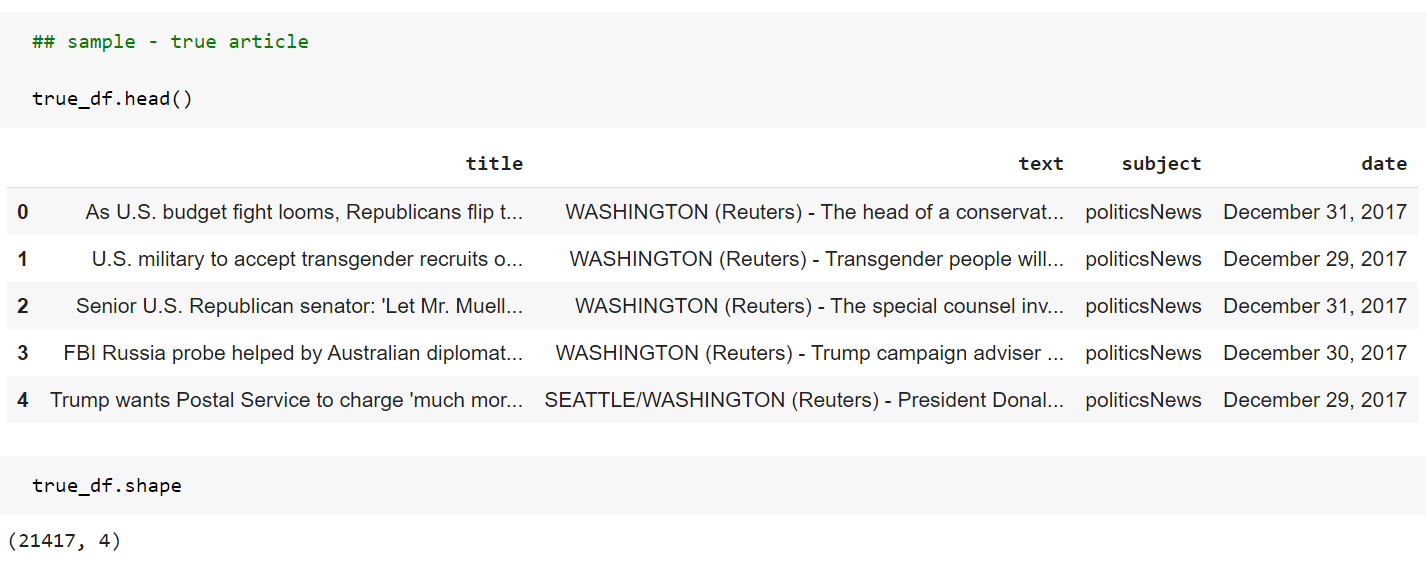


Figure 2. First 5 records in the true news articles dataset

1. **Data Preprocessing**

Data can contain missing values, unsuitable format or noise because they are gathered from various real-world sources. Data preprocessing is a very important task which requires cleaning the raw data obtained from various data sources and to make it suitable for machine learning models. Cleaning the data and converting it to the acceptable format is a very crucial and first step in doing any machine learning task. In the Kaggle dataset used for this work, there are 23481 fake news articles and 21417 true news articles. The following steps are done as part of data preprocessing:

1. **Create a new column ‘label’ :**

A new column **‘label’** is created in both the true dataset: **true\_df** ; and fake dataset: **fake\_df** which denotes the class of the news - whether the article is fake or true. The ‘label’ column is assigned – ‘1’ for true news article, ‘0’ for fake news article.

1. **Combine both the true\_df and fake\_df datasets:**

For the purpose of classification task, both the fake and true datasets are combined and shuffled to create the main dataset that would be used for training the model. The combined dataset is named **‘data’** and has **44898 records**. The resulting dataset will be balanced because it contains approximately equal number of fake and true news. A dataset that has a balanced number of records for both the classes will help to create a model with better accuracy.

1. **Create a new column ‘label’:**

The dataset **‘data’** is checked for any missing or NaN values. There are no missing values in the dataset. In order to avoid any errors, it is important to check for such missing values and removing them from the dataset before applying other data processing steps on the text data.

1. **Data Visualization**

1. **True and Fake news Distribution:​**

The distribution of True and Fake news articles is depicted using Pie charts to show the percentage of both the news in the dataset. To generate the charts, plotly python library is used. 47.7% of total news is true articles and 52.3% is fake articles.

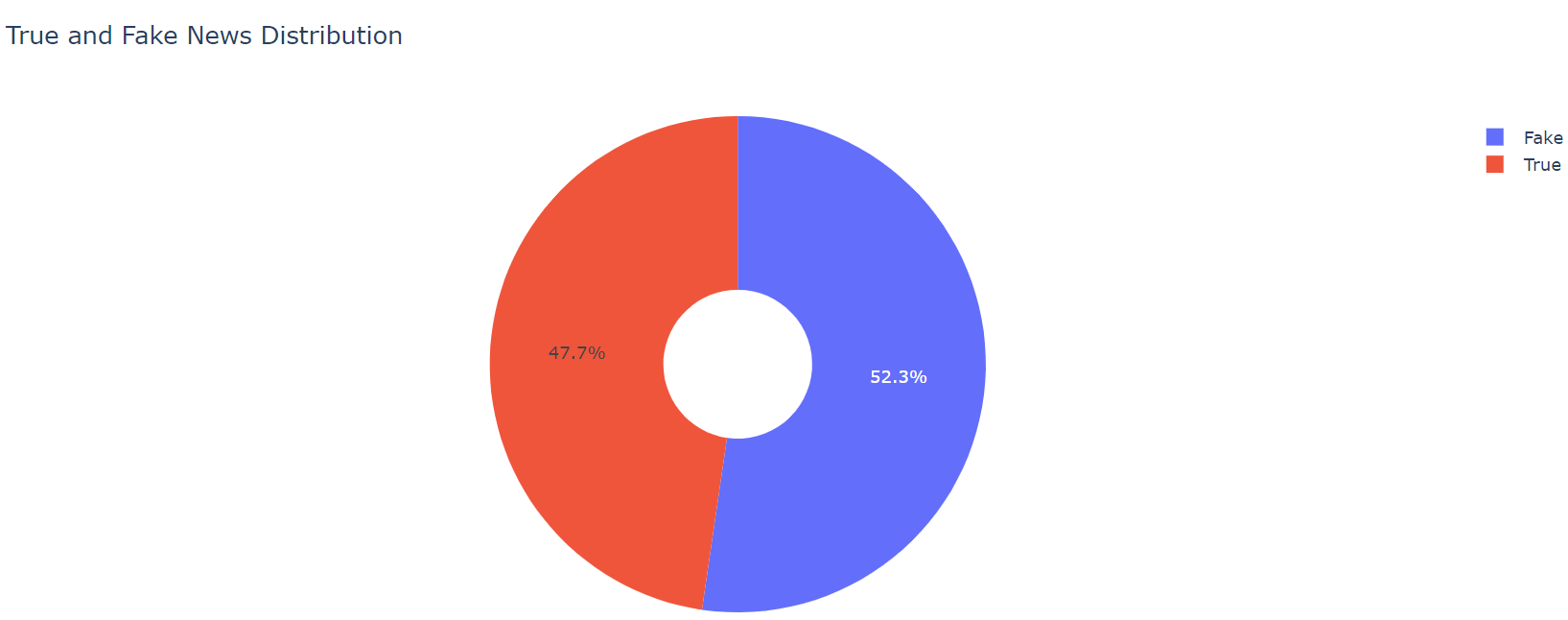


Figure 3. Pie-chart depicting distribution between fake and true news

**2. Subject wise news Distribution:​**

The news articles in the dataset belong to different subject categories. The subject categories are: Left News, World News, Politics News, Government News, Middle-east News, US News, etc. A bar-chart can be generated to view the number of articles belonging to each category as below:

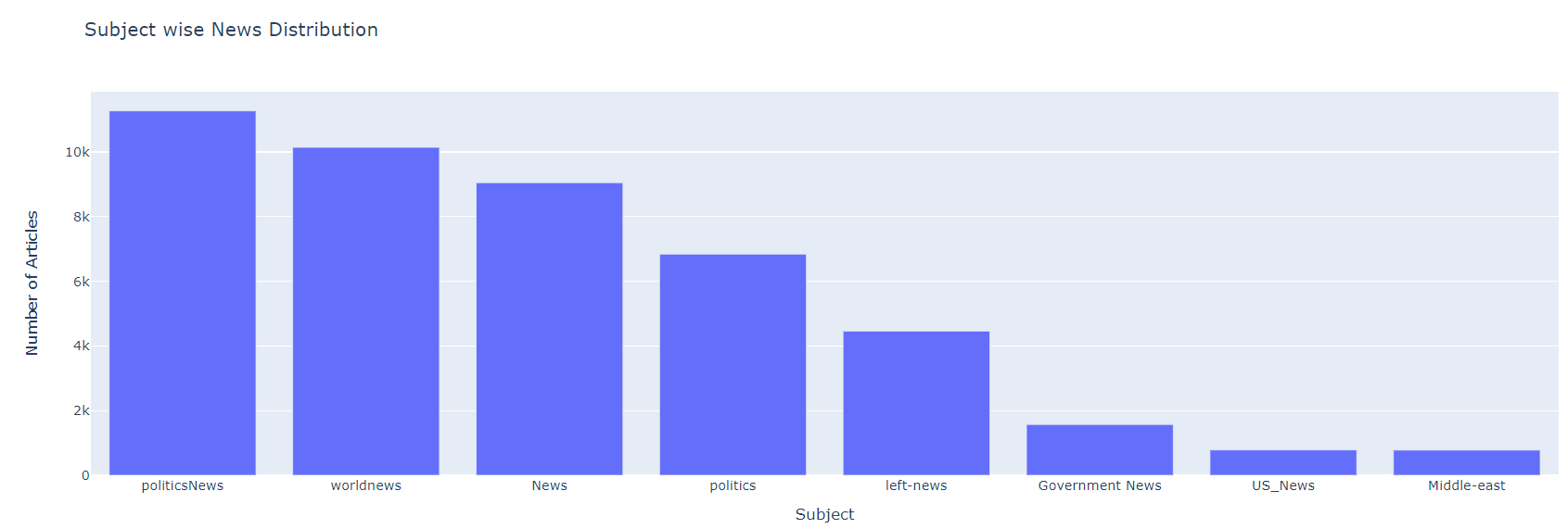


Figure 4. Bar-chart depicting subject wise news distribution

1. **Feature Engineering**

The data in the Kaggle datasets needs to be converted into proper data that can be given as input to the various machine learning models. Since each type of data needs a specific set of tasks to convert raw data into machine interpretable format, following steps are applied to convert text news data into features that are used as input to the models.

1. **Dependent and independent features:​**

For the fake news classification task, the **‘text’** column will be used to train the model and predict the **‘label’** to which it belongs. Therefore, the **‘text’** column values will be used to extract independent features and the **‘label’** column becomes the dependent feature**.**

1. **NLTK to clean text:​**

NLTK library from python is very powerful and useful for any Natural Language Processing Task. With the help of NLTK stopwords and Porter Stemmer, the news article text data can be cleaned in order to reduce the text length and get more meaningful words. Both these techniques are applied on the **‘text’** column.

**Stopwords:** Stopwords are the most common words in any natural language. For the purpose of analyzing text data and building NLP models, these stopwords might not add much value to the meaning of the document. Generally, the most common words used in a text are “the”, “is”, “in”, “for”, “where”, “when”, “to”, “at” etc.

For text classification, where the text is to be classified into different categories, stopwords are removed or excluded from the given text so that more focus can be given to those words which define the meaning of the text.

**PorterStemmer:** Stemming is normalization of words, which means reducing a word to its root form. Stemming cuts off the end or beginning of a word by taking into account a list of common prefixes or suffixes that could be found in that word. In most natural languages, a root word can have many variants. For example, the word ‘play’ can be used as ‘playing’, ‘played’, ‘plays’, etc.

1. **TF-IDF Vectorizer:​**

The TfidfVectorizer class from the sklearn.feature\_extraction.text module can be used to create feature vectors containing TF-IDF values. TF-IDF is a product of two terms: TF(Term Frequency) and IDF(Inverse Document Frequency). TF-IDF which measures how important a particular word is with respect to a document and the entire corpus. Words which are rare in a document will have a high score in the TF-IDF vector.

An N-Gram is a sequence of N-words in a sentence. Here, N is an integer which stands for the number of words in the sequence. For example, N=1, then it is referred to as a uni-gram. If N=2, then it is a bi-gram. Similarly if N=3, then it is a tri-gram.

Here, 5000 features are created using TFIDF Vectorizer with N-gram range (1,3). These 5000 most relevant features in the corpus are selected to be used for training the model.



Figure 5. First 10 features from the TF-IDF Vectorizer

All the news articles in the dataset will be now represented by these 5000 features and each feature will have a value depending upon its occurrence in the article.

1. **Split Train and Test data:​**

The dataset is divided into Train and Test data. 70% of the data will be used to train the model and 30% of the data will be used to predict the output and test the model for accuracy. The random state is set as 0 during the split process. The train dataset is used to train the classifier while building the model. To check how well the model has learned during the training, testing data is given input to the model to generate predictions. The efficiency of the classifier is observed from the accuracy score.

1. **Machine Learning Models**

Machine learning algorithms have proved to be very useful when solving complicated tasks such as face recognition, speech identification, price prediction, etc. These algorithms can also give promising results when applied to text classifications tasks such as fake news classification. This works aims to solve this task with the help of various machine learning models and then compare the performances to see which one works best with the given dataset.

The news articles (text data) is first preprocessed to obtain clean text (by removing stopwords and applying stemming). Then this clean text data is converted into features which are vector representation of the textual data. The features are obtained using TF-IDF vectorizer. The complete dataset is divided into train & test dataset. The train dataset is used to train the classifier and test dataset is used to measure the performance of the classifier. The features from the vectorizer are given as input to the various machine learning models. The model is tested on the train dataset and accuracy is obtained for each machine learning model. The different models used are discussed in the subsequent section. The overview of the complete process is represented by the following diagram:

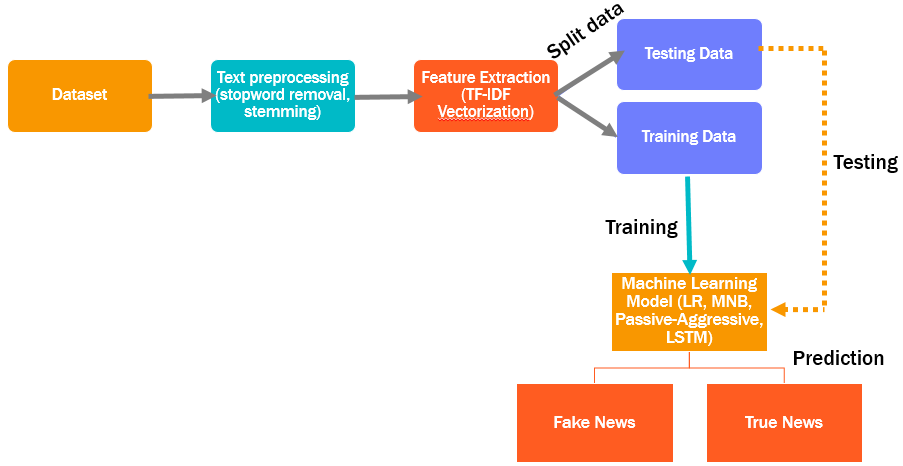


Figure 6. Overview of the processed workflow

* 1. Multinomial Naïve Bayes Algorithm:​

The Multinomial Naive Bayes algorithm is based on the Bayes theorem. It calculates the probability of an event occurring based on the prior knowledge of conditions related to an event, based on the following formula:

P(A|B) = P(A) \* P(B|A)/P(B)

The probability of event A is calculated when probability of event B is already provided.

P(B) = prior probability of B

P(A) = prior probability of A

P(B|A) = occurrence of B given probability of A

The above formula calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

A Naive Bayes classifier is a probabilistic machine learning model that is used for classification task. It is fast and easy to implement and one of the most popular and simple machine learning classification algorithms.



Figure 7. Confusion Matrix for Multinomial Naive Bayes Classifier

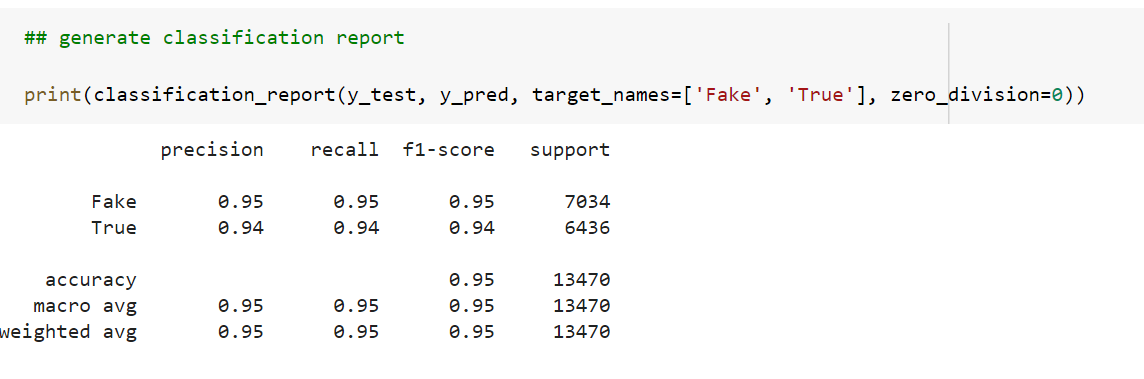


Figure 8. Classification report for Multinomial Naive Bayes Classifier

* 1. Logistic Regression Algorithm:​

Logistic regression is a supervised classification algorithm, used when the value of the target variable is categorical in nature. The target variable (or output), y, can take only discrete values for given set of features (or inputs), X. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. It is a powerful machine learning algorithm that utilizes a sigmoid function and works best on both binary and multi-class classification problems.

Logistic regression becomes a classification technique only when a decision threshold is set. This setting is dependent on the classification problem itself. The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but this doesn’t happen practically.

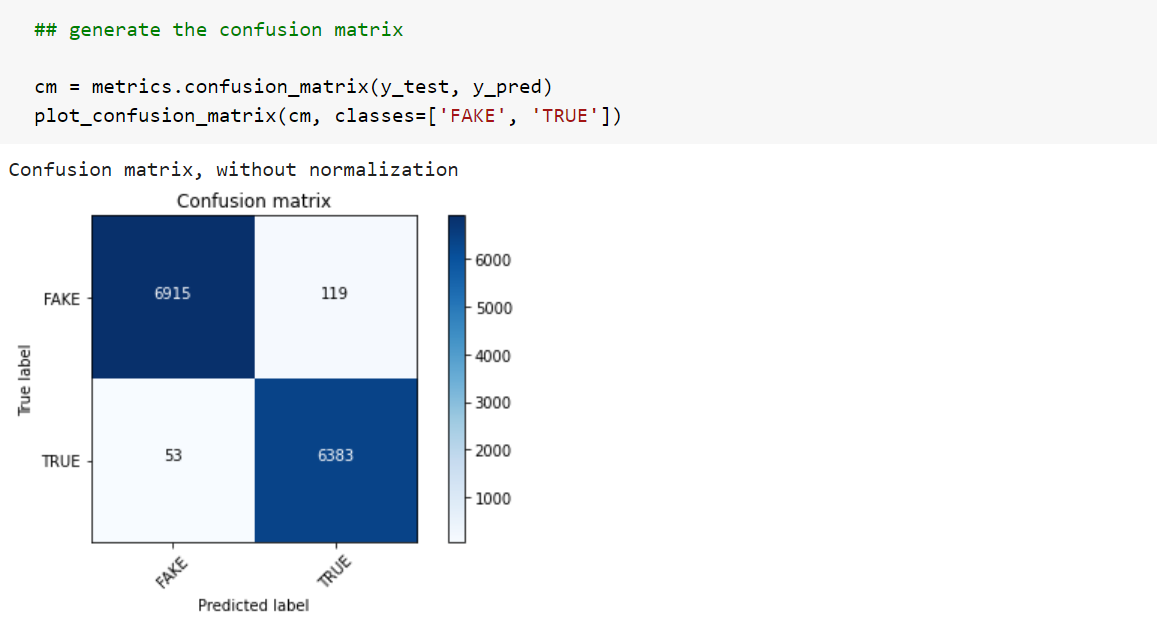


Figure 9. Confusion matrix for Logistic Regression Classifier

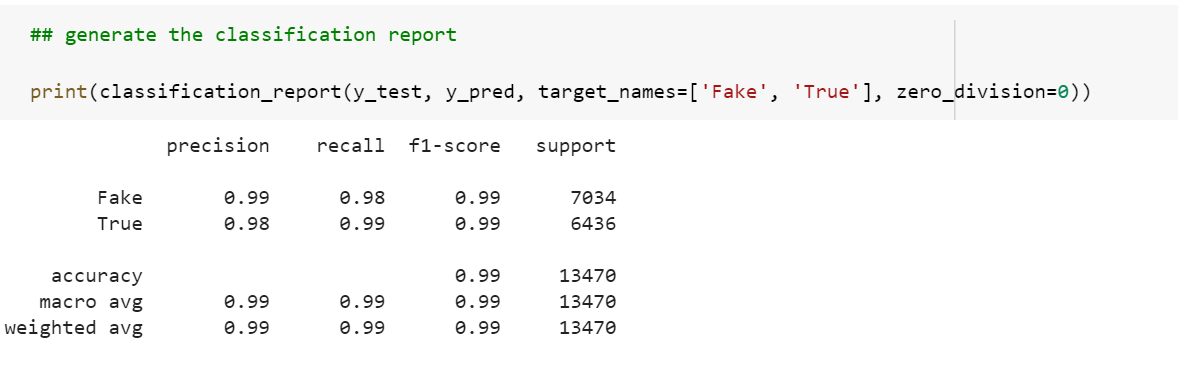


Figure 10. Classification report for Logistic Regression Classifier

* 1. Passive Aggressive Classifier Algorithm:​

The Passive Aggressive Classifier Algorithm is an “online learning algorithm”. These algorithms are generally used for large-scale learning. The algorithm remains passive if it achieves correct result after classification, and gets aggressive if there is error in calculation or result. Because of this approach, the algorithm doesn’t converges. It uses marginal values to correct the current classification task.

In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. This is very useful in situations where there is a huge amount of data and it is computationally infeasible to train the entire dataset because of the sheer size of the data.

Passive-Aggressive algorithms do not require a learning rate. However, they do include a regularization parameter. Passive-Aggressive algorithms are called so because- *Passive*: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.  *Aggressive*: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

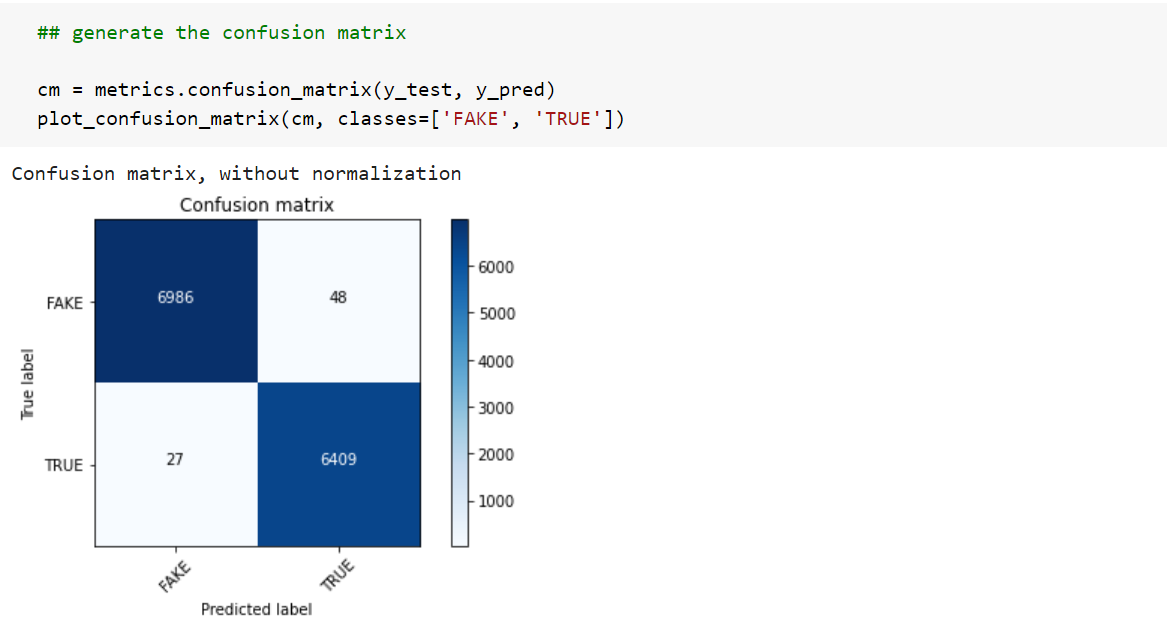


Figure 11. Confusion matrix for Passive Aggressive Classifier

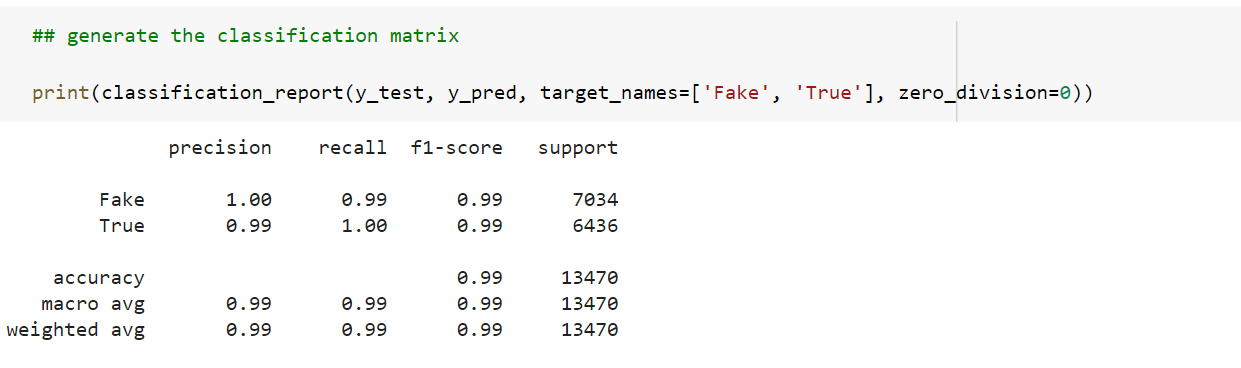


Figure 12. Classification report for Passive Aggressive Classifier

In this fake news classification task, Passive-Aggressive Classifier gives the best performance. It gives both accuracy and F1-score of 99%, the classifier can be used to predict the most relevant features for this dataset.

* 1. LSTM (Long short-term memory) Algorithm:​

The Long short-term memory (LSTM model) is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data.

The reason to use LSTM is that it is effective in memorizing important information. In non-neural network classification techniques, models are trained on multiple word as separate inputs that are just word having no actual meaning as a sentence, and while predicting the class it will give the output according to statistics and not according to meaning. That means, every single word is classified into one of the categories.

However, LSTM uses a multiple word string to find out the class to which it belongs. This is very helpful while working with Natural language processing. Using appropriate layers of embedding and encoding in LSTM, the model will be able to find out the actual meaning in input string and will give the most accurate output class. In this task, the LSTM model has one embedding layer, one LSTM layer and one dense layer. The loss function used is *binary\_crossentropy* and *ADAM* optimizer is used. The LSTM model achieves an accuracy score of 95.2% and F1-score of 94.9%.

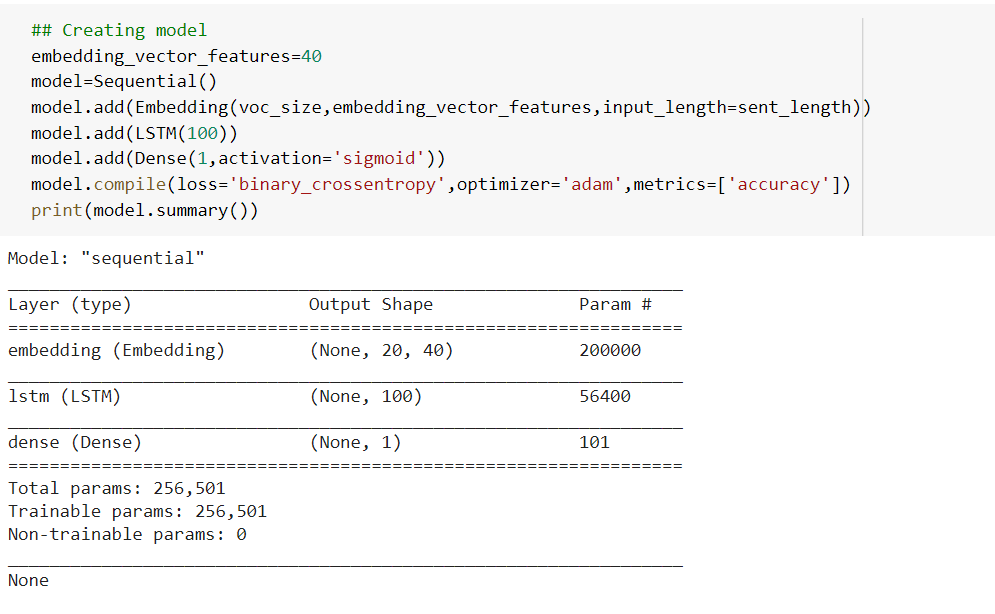


Figure 13. Model Summary for LSTM neural network model

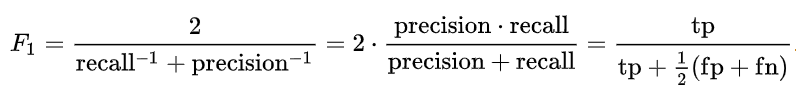
1. **Results**

To evaluate the performance of the different machine learning models used in this fake news classification task, performance metrics used are accuracy and F1-score. F1-score is harmonic mean of recall and precision values. Precision is positive predictive value and recall is the sensitivity value. The precision and recall of a classifier is calculated as follows:

Precision = tp / (tp + fp), Recall = tp / (tp + fn)

Where tp – number of true positive examples; fp – number of false positive examples; fn is a number of false negative examples

The F1 score is calculated as:



The Multinomial Naïve Bayes classifier gives an accuracy score of 94.6% and F1-score of 94.4%. Logistic Regression model achieves an accuracy score of 98.7% and F1-score of 98.6%. The neural network based LSTM model also gives a decent performance with an accuracy score of 95.2% and F1-score of 94.9%. In this fake news classification task, Passive-Aggressive Classifier gives the best performance. It gives both accuracy and F1-score of 99.4%.

1. **Conclusion**

Fake news detection is a challenging task and is becoming more crucial than ever before because of the huge amount of misinformation spreading rapidly like a wildfire. This work aims to classify fake news by applying some machine learning models on a sample dataset. The models explored are Logistic Regression, Multinomial Naïve Bayes, Passive Aggressive classifier and LSTM neural network model. This study shows that Passive Aggressive classifier can give good results on fake news classification task. Such a system can be quite useful for the social media users and journalists. This research area has a lot of potential to be explored.

For future work, the analysis done in this work can be extended and improved in many ways. Word-embedding like Word2Vec and GloVe can be used to obtain more meaningful features from the text data. Novel machine learning architectures like convolution neural network for text and transformer based text classification methods can be used to enhance the performance of the system. Also, by increasing the size of the dataset better results can be obtained. A system trained on a larger set of data will be more robust and have good accuracy. With the help of artificial intelligence, the spread of fake news can be controlled more quickly and efficiently as compared to manual efforts.

1. **References**

[1] G. staff and agencies, “Washington gunman motivated by fake news ‘pizzagate’ conspiracy,” 12 2016. [Online] Available: <https://www.theguardian.com/us-news/2016/dec/05/gunman-detained-atcomet-pizza-restaurant-was-self-investigating-fake-news-reports>.

[2] J. Zarocostas, “How to fight an infodemic,” The Lancet, vol. 395, no. 10225, p. 676, 2020. [Online] Available: <http://www.sciencedirect.com/science/article/pii/S014067362030461X>.

[3] Buntain, Cody, and Jennifer Golbeck. "Automatically identifying fake news in popular twitter threads." 2017 IEEE International Conference on Smart Cloud (SmartCloud). IEEE, 2017.

[4] Granik, Mykhailo, and Volodymyr Mesyura. "Fake news detection using naive Bayes classifier." 2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON). IEEE, 2017.

[5] Kaliyar, Rohit Kumar. "Fake news detection using a deep neural network." 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.

[6] Amine, Belhakimi Mohamed, Ahlem Drif, and Silvia Giordano. "Merging deep learning model for fake news detection." 2019 International Conference on Advanced Electrical Engineering (ICAEE). IEEE, 2019.

[7] Yanagi, Yuta, et al. "Fake news detection with generated comments for news articles." 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES). IEEE, 2020.

[8] Shu, Kai, et al. "Fake news detection on social media: A data mining perspective." ACM SIGKDD explorations newsletter 19.1 (2017): 22-36.

[9] Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detection of online fake news using n-gram analysis and machine learning techniques." International conference on intelligent, secure, and dependable systems in distributed and cloud environments. Springer, Cham, 2017.

[10] Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detecting opinion spams and fake news using text classification." Security and Privacy 1.1 (2018): e9.

[11] Traylor, Terry, Jeremy Straub, and Nicholas Snell. "Classifying fake news articles using natural language processing to identify in-article attribution as a supervised learning estimator." 2019 IEEE 13th International Conference on Semantic Computing (ICSC). IEEE, 2019.

[12] scikit-learn. 2021. 1.9. Naive Bayes. [Online] Available at: <https://scikit-learn.org/stable/modules/naive_bayes.html>.

[13] Kaggle.com. 2021. Fake and real news dataset. [Online] Available at: <https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset>.

[14] scikit-learn. 2021. sklearn.linear\_model.PassiveAggressiveClassifier. [Online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PassiveAggressiveClassifier.html>.