

# Fake News Detection

Course Title: Machine Learning

Course code: CSE475

Section: 02

Semester: SUMMER'21

SUBMITTED BY,

Maria Mehjabin Shenjuti (2018-1-60-244)

MD.Muhaimenul Islam(2018-1-60-017)

Nazifa Tabassum(2018-1-60-232)

Mohammad Sadat Hossain(2018-1-60-096)

Jasmine Akter Ratri(2018-1-60-074)

Date Of Submission: 13/09/2021

# 1. Introduction

Fake news exists way before from social media, but it multifold when social media was introduced. Fake news is a news designed to deliberately spread hoaxes, propaganda and disinformation. Fake news stories usually spread through social media sites like Facebook, Twitter etc.

Social network sites are the common platforms that allow the user to create and share the content or to take an interest in social networking. But the spreading of wrong information creates a negative effect on our community which we called it a fake news.

Types of fake news: Visual based type: Visual based are mainly photoshopped images and videos which are posted in social medias. Linguistic based type: Linguistic based are mainly the manipulation of text and string content. This issue is with blogs, news, or emails. In our project, the dataset will be in the form of text or news article. That news article will be labelled such as fake or not fake.

## 1.1 Objectives

Regarding this problem some objectives are: a) To develop a clean dataset based on the context of social media. b) To recognized detect the fake news using ML. c) To evaluate the effectiveness, accuracy of the algorithm and the credibility of the news. d) The project aims to develop a method for detecting and classifying fake news stories using Naive Bays, SVM, RNN, Logistic Regression. e) The main goal is to identify fake news, which is a classic text classification issue. f) We gathered our data, preprocessed the text, and translated our articles into supervised model features. g) Our goal is to develop a model that classifies a given news article as either fake or true.

## 1.2 Motivation

Fake news is a widespread problem and due to the digitalization tackling it is another challenge. Fake news can propagate through the thousands of the digital source available nowadays. Ai bots are also used for the spread of fake news. And people tend to believe on anything that they read on the

internet and those who are new to this technology era are easily fooled. Mis-information can cause crime and destruction of society that is why detecting and stopping fake news is necessary.[1]

## 1.3 Existing Works

There are numbers of research there for detecting the fake news. Everyone has their own way of making the dataset and algorithm for detecting the fake news and they have their own accuracy. Data mining is used to find the fake news. Feature extraction along with metric evaluation is used in a methodology to find the fake news. Bot spamming, click bait, source of news also effected the process of fake news detection[2]. Researchers also used neural-network-based AI as they developed. RNN was also used to detect the fake news. RNN was used to detect temporal pattern of user activity around a specific article and then source character and finally classifier for fake news detection[3]. Another research used CNN to classify the fake news which was infused with LSTM. For this GloVe library was used[4]. Till now many approaches has been taken to find the fake news and it has various accuracy. So, there is scope to work in this field and find some new path.

## 1.4 Necessity

After collecting the dataset fake and real news can be determined with highest accuracy which will save the people from consuming the fake news. It will stop the propagation of the fake news and different platform can integrate this also to find the fake news and stop it from spreading

## 2. Methodology

Algorithm: We used four different classifiers for our project. They are:

Naive Bayes is a probabilistic that applies Bayes theorem.

A support vector machine or simply SVM is discriminating classifier formally defined by a separating hyperplane.

Recurrent Neural Network or simply RNN is a type of artificial neural network commonly used in speech recognition and natural language processing.

Logistic regression is statistical model that uses a logistic function to model

a binary dependent variable.

Data manipulation using pandas (It refers to the various processes such as data loading exploring, cleaning, transforming etc.)

Natural language processing (NLP)

Stemming

Stop-words

N-grams

Tokenization

Vectorization

Bag of words.

The various NLP techniques such as stemming, removing, stop-words, n-grams are implemented as the news input is in a natural language and the machine needs these techniques to process and analyze them.

For detecting the fake news, we can follow the following steps-

- Collect the raw dataset.
- Apply preprocessing and clean the data by using NLP techniques.
- Apply NLP model to convert the text data into vector form.
- Apply machine learning models to classify the articles as a fake or not fake.
- Evaluate the results.
- Visualize the result.

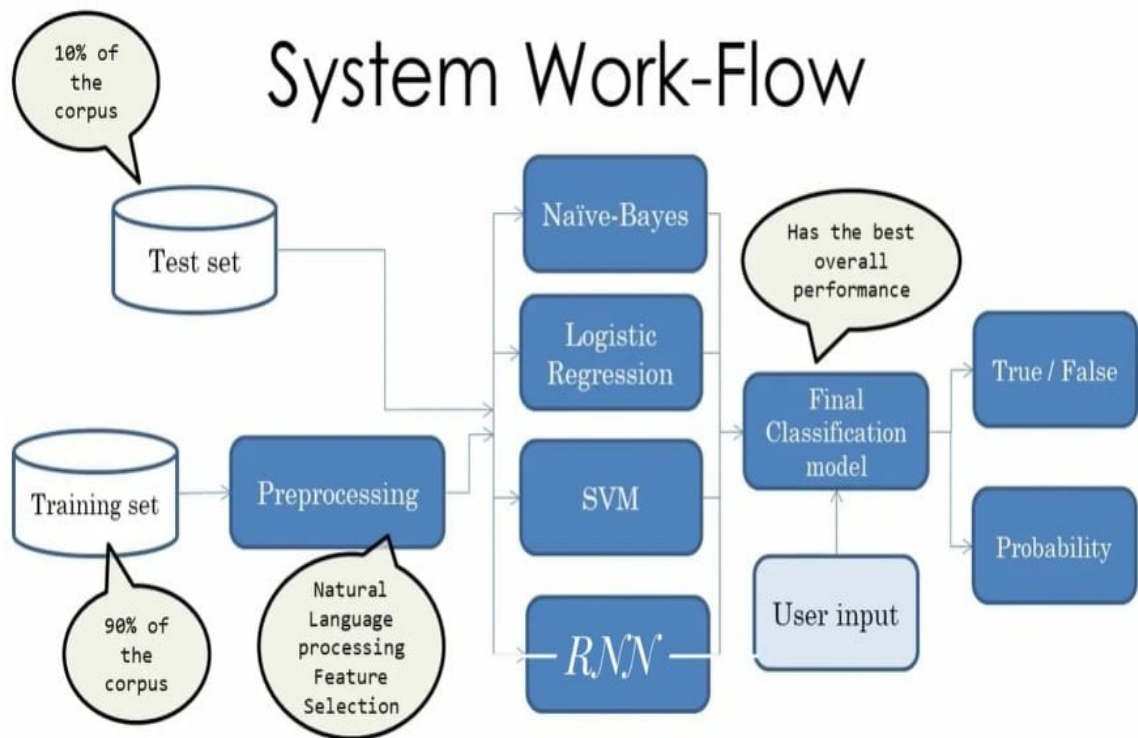


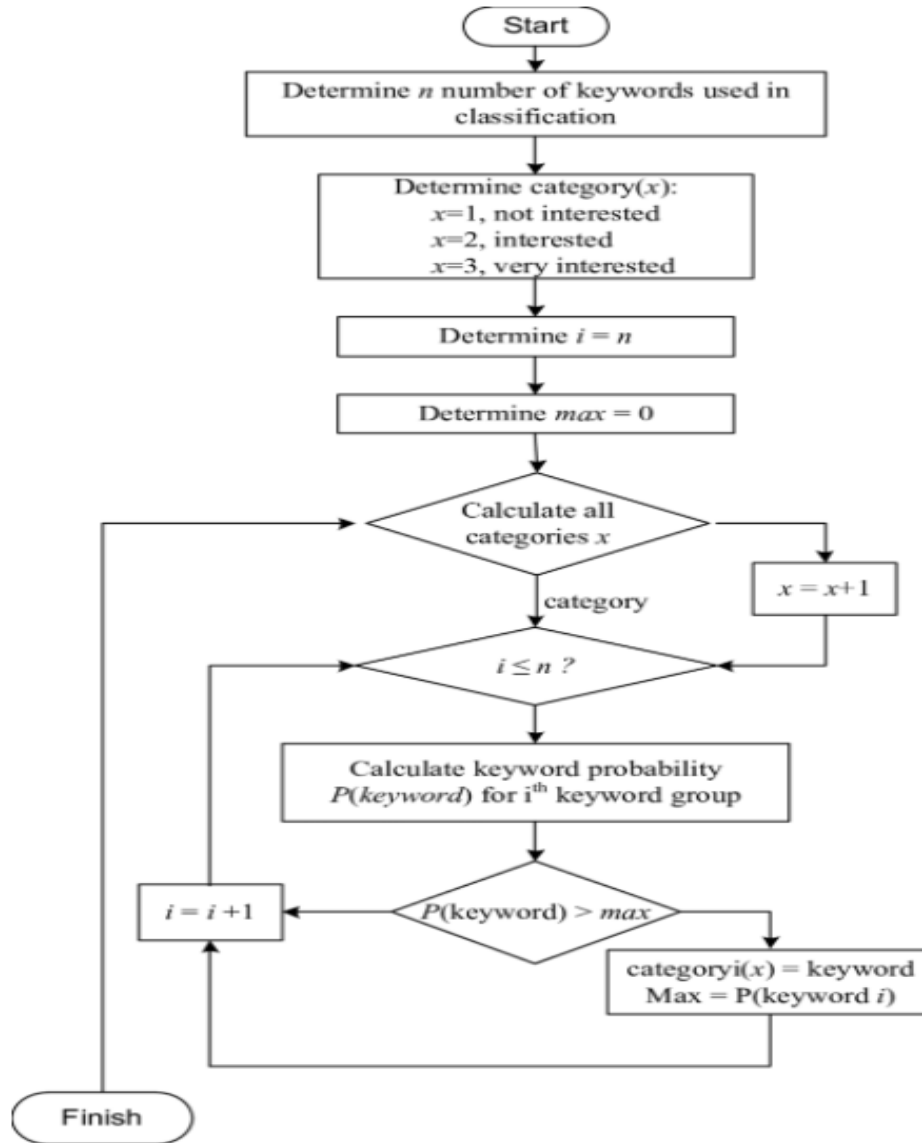
Figure: Work Flow

## Naïve Bayes:

Naïve Bayes is a conditional probability model which can be used for labeling. The goal is to find a way to predict the class variable (B) using a vector of independent variables (A), i.e., finding the function  $f: A \rightarrow B$ . In probability terms, the goal is to find  $P(B|A)$ , i.e., the probability of B belonging to a certain class A. B is generally assumed to be a categorical variable with two or more discrete values. It is a mathematically simple way to include contributions of many factors in predicting the class of the next data instance in the testing set. The limitation of Naïve Bayes is that they assume that all features are not dependent on each other. The Naïve Bayes rule is based on the theorem formulated by Bayes:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

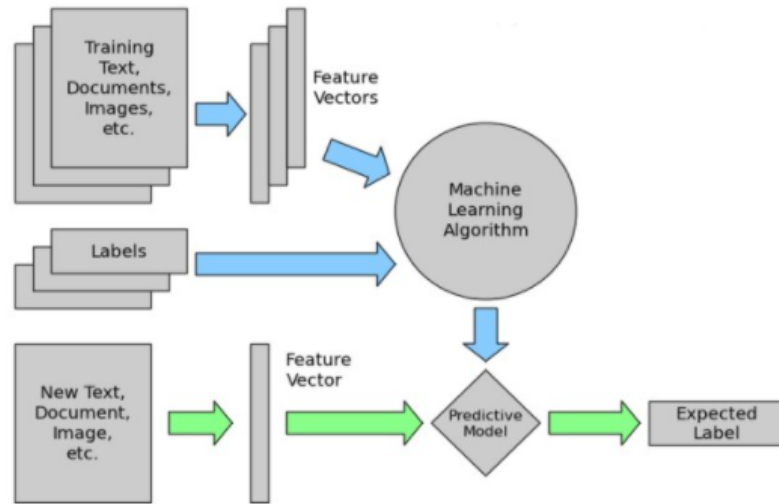
## Working Method:



Naive Bayes algorithm flowchart

## Text Classification Model:

The general workflow for this task can be described as,



[5]

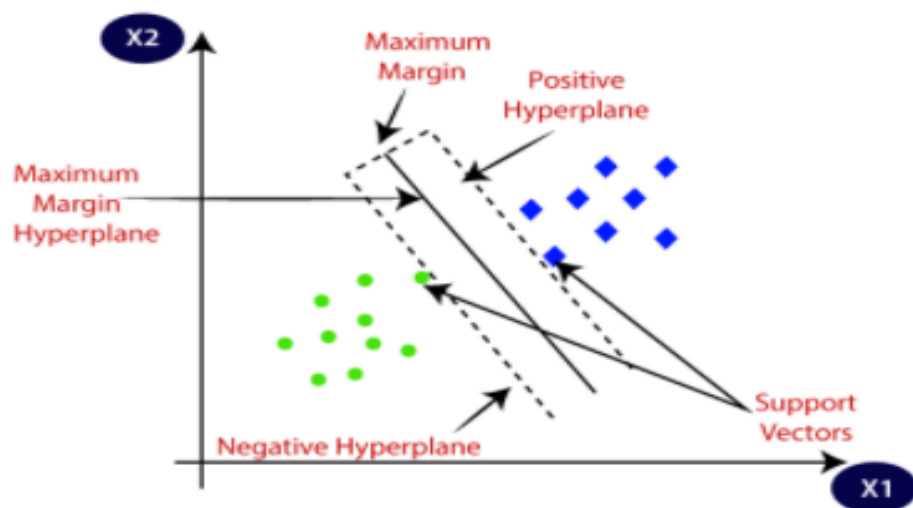
### A support vector machine or simply SVM:

A support vector machine (SVM), which can be used interchangeably with a support vector network (SVN), is also considered to be a supervised learning algorithm. SVMs work by being trained with specific data already organized into two different categories. Hence, the model is constructed after it has already been trained. Furthermore, the goal of the SVM method is to distinguish which category any new data falls under, in addition, it must also maximize the margin between the two classes. The optimal goal is that the SVM will find a hyperplane that divides the dataset into two groups. The kernel used in this application is RBF as it is best suited for large applications like a corpus of news articles. The Radial Basis function on two samples  $x$  and  $x'$  is given by:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

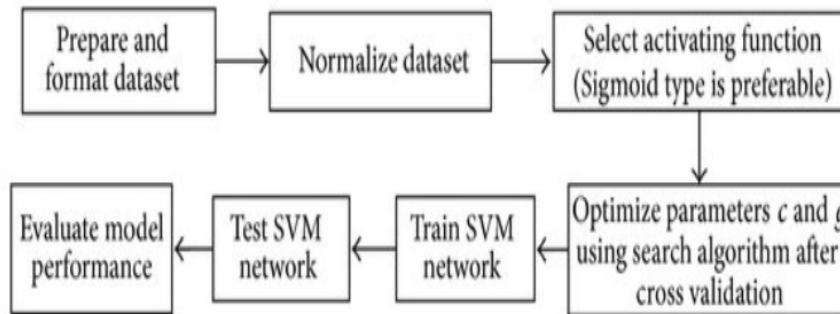
Where  $\|x - x'\|^2$  represents the squared Euclidean distance and  $\sigma$  is a free parameter.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:





### Working Method:

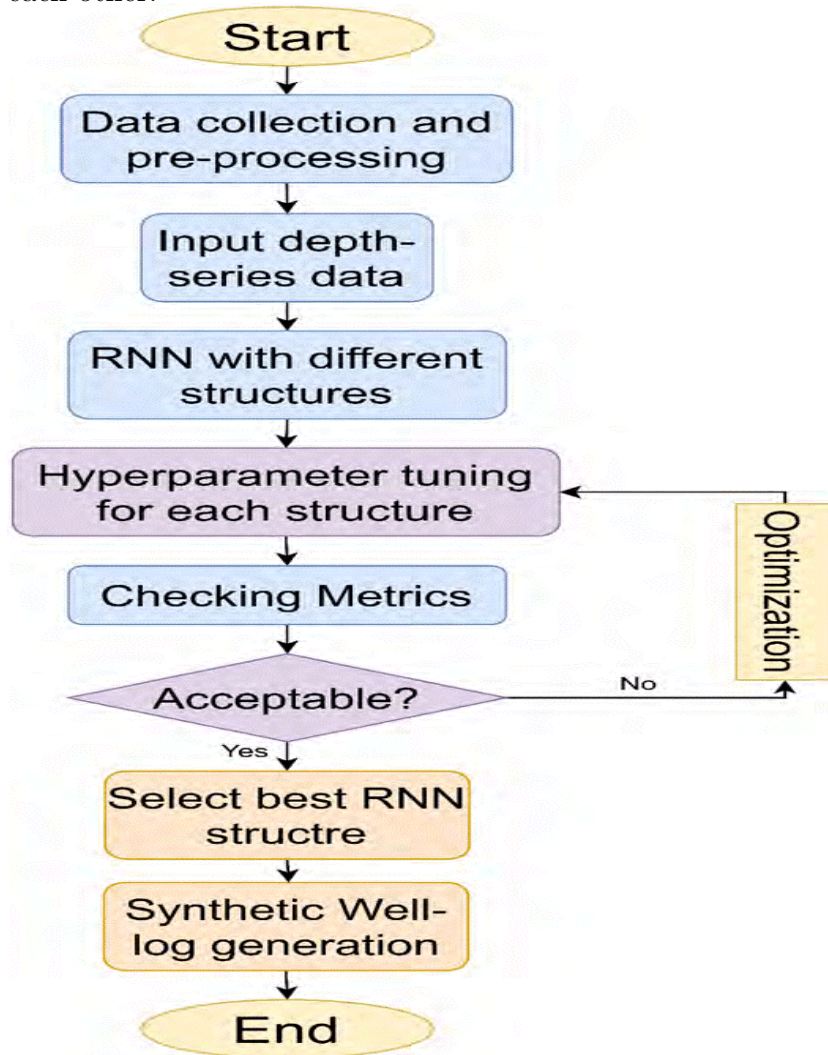


[6]

### Recurrent Neural Network or simply RNN:

Text comes in various formats. The text of Twitter tweets can be really messy. News articles can use a lot of blank lines. The first step in every Natural Language Processing task is to clean the text, by removing blank lines and unnecessary punctuation. The next step is to transform the words to numbers, because computers can't read words. There are multiple ways of doing this. The easiest way is to assign numbers for each unique word, and then use that as input for a model. So, the sentence "Last summer the singer toured the Westcoast of the USA" would become [1 2 3 4 5 3 6 7 3 8]. As you can see each unique word is assigned a new number. Another more common method is to use pretrained word embeddings. Instead of converting every word in a number, words are converted to a tensor representation. So when a 4-dimensional embedding is used, every (unique) word is converted to a combination of four numbers. Word embedding work so well because the semantics of the words are captured, words with the same meaning have similar tensor values and differences with other word groups are similar as well. The example underneath visualizes this. As you can see the words all get bigger Y values when the word is feminine. After the words are converted into word embeddings, the words are fed into a neural network. This neural network consists of various layers. The first layer is a convolutional layer. A convolution is a filter that can extract features from the data. In image detection for example, convolutions can be used to detect edges or shapes. In Natural Language Processing convolutions can also improve performance. The example below shows a filter that extracts relation between two words that have one word in between. In each step, the filter is multiplied with

the word embedding values. The filter values of 1 time the word embedding values result in the word embedding values, while filter values of 0 result in 0. The next layer is a max pooling layer. This layer iterates over the tensors and takes the highest value. In this way the feature space is compressed. This step makes sure important features are kept, while empty space is dropped. The next layer is a Long Short-Term Memory (LSTM) Layer. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The last layer before the prediction is made is a fully connected layer. This is a regular neural network layer where all nodes are connected with each other.

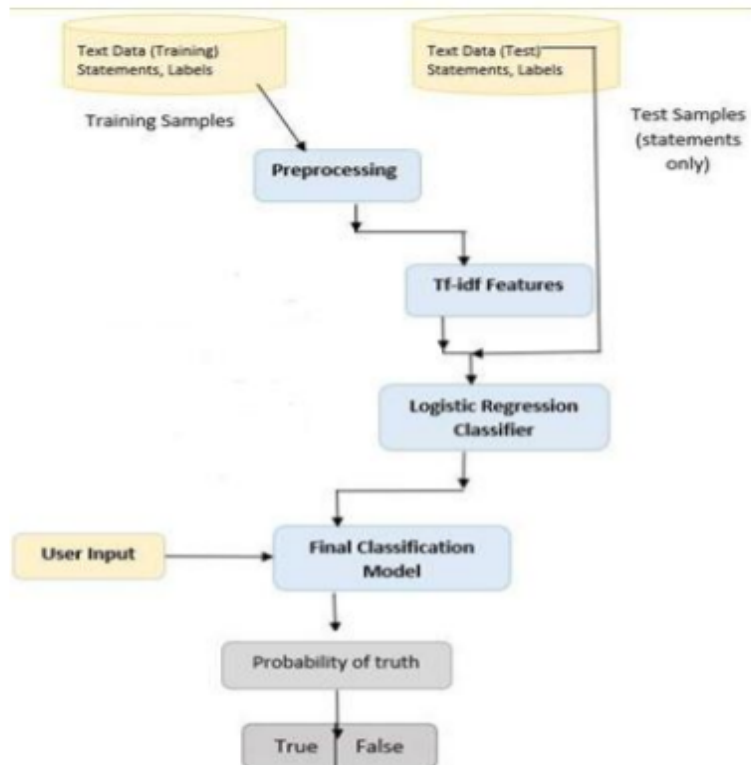


[7]

### Logistic regression:

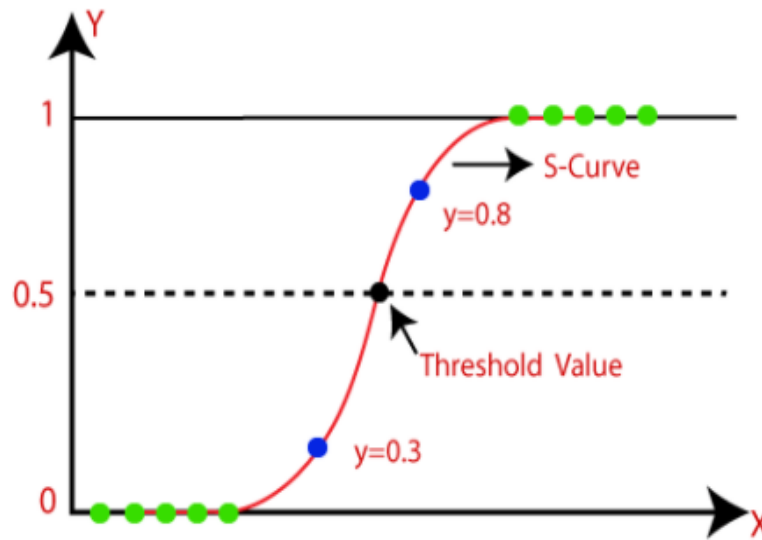
Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc. Logistic regression is a simple yet very effective classification algorithm so it is commonly used for many binary classification tasks. Logistic regression model takes a linear equation as input and use logistic function and log odds to perform a binary classification task. Logistic regression predicts the output of a categorical dependent variable.

### Working Method:



Below are the steps: Data Pre-processing step. Fitting Logistic Regression to the Training set. Predicting the test result. Test accuracy of the result (Creation of Confusion matrix) Visualizing the test set result.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



Logistic regression Classifier is a ML classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts  $P(Y=1)$  as a function of  $X$ .

### 3. Implementation

At first, we have collected a dataset for each of our algorithm which was Naïve Bayes, SVM, RNN and Logistic Regression. After that we have to pre-process each of those datasets for a better result as it might contain irrelevant stuffs that might decrease the accuracy of the algorithms. We have visualized our distribution of fake news and real news in bar chart for a better understanding. Then we have applied the Naïve Bayes, SVM, RNN and Logistic Regression algorithms to find our results. We have calculated

results such as accuracy, precision, f1-score, etc. and plotted all those results in a bar chart for a clear view so that it is easier to compare our results and find out which algorithm has a better accuracy.

## 3.1 Data Collection

- Naïve Bayes and SVM: The dataset that was used is Kaggle's 'Fake News Detection' and was taken from the following link: <https://www.kaggle.com/jruvika/fake-news-detection>

This dataset contains the following properties and they are:

- URL: This column contains the link of every news
- Headline: This column contains the headline of each news
- Body: This column contains the detailed explanation of every news
- Label: This column contains the type of each news

This dataset contains a total of 4009 news articles.

- RNN: For this algorithm, we have collected the data sample from the following link: <https://www.kaggle.com/clmentbisailon/fake-and-real-news-dataset>

This dataset is composed of 3 features and they are:

- Title: This column contains the titles of every news
- Text: This column contains the text description of each news title
- Subject: This column contains the subject of each news

Total number of samples in this dataset is 44898.

- Logistic Regression: The dataset used for this algorithm was collected from the internet.

## 3.2 Data Processing

### Naïve Bayes and SVM

After the data was read, the pre-processing tasks were performed on this dataset to remove all the unnecessary things. At first, we have removed all the punctuations and special characters from our dataset. After that we have to convert all the articles into lowercase letters but before doing that, we have tokenized the article. Then we have converted all the articles into lowercase letters and also removed all the stop-words. Stop-words constituted unnecessary part of the news articles and it does not indicate whether a news

article is real or not. That is why stop-words were removed. After that we have used TF-IDF and Bag of words for text vectorization.

### **RNN**

After we have read the data, we have checked whether any dataset contains any null value or not. Then we have checked for the unique values for the subject part of our dataset. Then from our dataset we have dropped few columns like date and subject as there are no strong relation between the date and the validity of the dataset. We have assigned some values for fake news and for real news like 0 for fake news and 1 for real news. Then we have normalized our data by removing lower case letters, extra spaces and url links. The data is also divided into training part and testing part and all of the sample are vectorized because our classifier only handles numerical data. We have also used padding so that all the articles have the same length.

### **Logistic Regression**

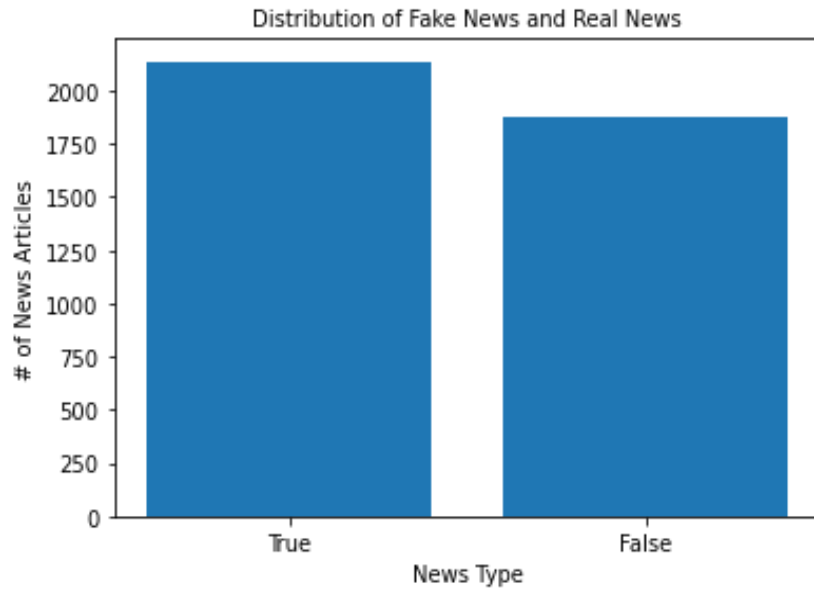
In the data set there are 586 news that are authentic and 578 news which are fake. Dataset is almost balanced here. For simplification title and text is merged to create titletext. Initially the dataset had html tags, punctuations and higher and lower-case data mixed. To make the data processing simple these tags are removed along with the punctuation and all the letters are converted to lower case. Later vectorizer is used to convert the whole title text dataset into titletext format using TFIDF. Dataset is split into train and test type and 50 percent of the data is used for training and other 50 percent for testing.

## **3.3 Model Development**

For our model development we have used Anaconda's Jupyter Notebook as our project environment and for the language, we have used python to develop our code.

## **3.4 Results**

**Naïve Bayes and SVM** At the very beginning of our implementation, we have made a bar chart to show the distribution of both true news and false news. This bar chart also helps us to identify the difference between these two types of news.



From the above figure, we can see that the proportion of real news in our dataset is greater than the proportion of false news. After the distribution, data was pre-processed for both Naïve bayes algorithm and SVM algorithm. Then the data set was divided into training set and testing set. When Naïve Bayes algorithm was used, it achieved an accuracy of 93.9percent for the test-ing dataset and for the training set, it achieved an accuracy of 97.4percent. The overall performance of Naïve Bayes is considered to be very good.



Figure: Accuracy of Naïve Bayes algorithm

**For SVM algorithm**, Stochastic Gradient Descent (SGD) is used for training the dataset. This is very efficient and is easy to implement. SVM with SGD training is implemented so that the gradient of the loss is estimated for each sample at a time and the model is updated along the way with a decreasing learning rate. After getting the best parameters, SVM showed an accuracy of 97.4percent on the testing dataset and 99.8percent for the training dataset.



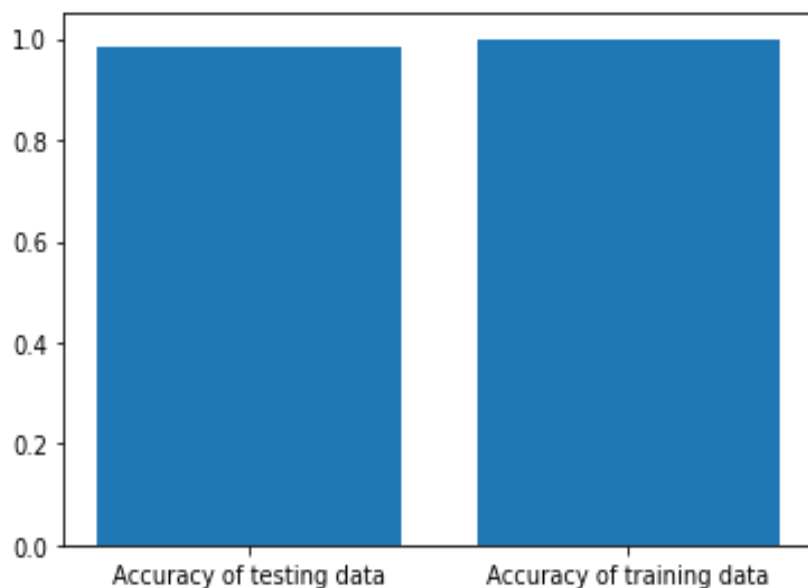


Figure: Accuracy of SVM algorithm

We also made a classification report that includes the result of precision, f1-score, support, etc. and plotted them in a graph. For simplification precision, recall and f1 score are shown in 100percent and support is as it was.

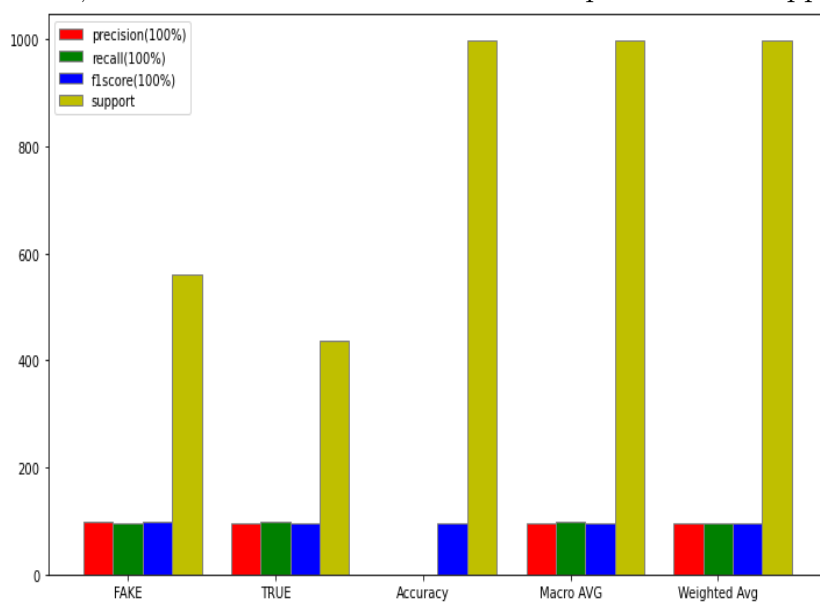
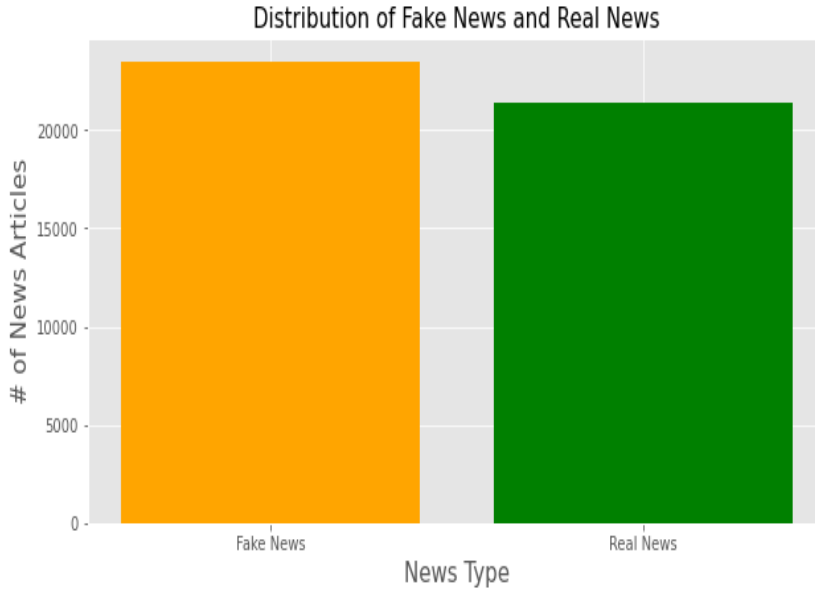


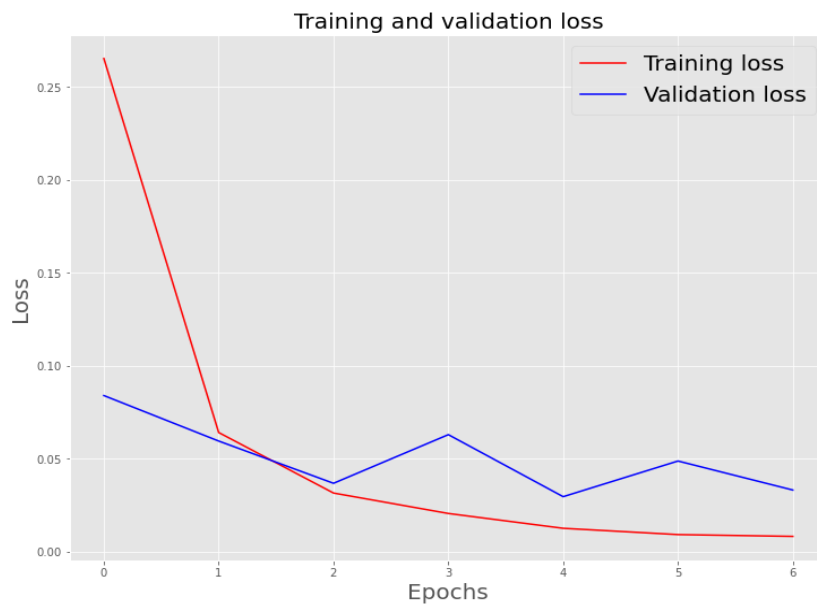
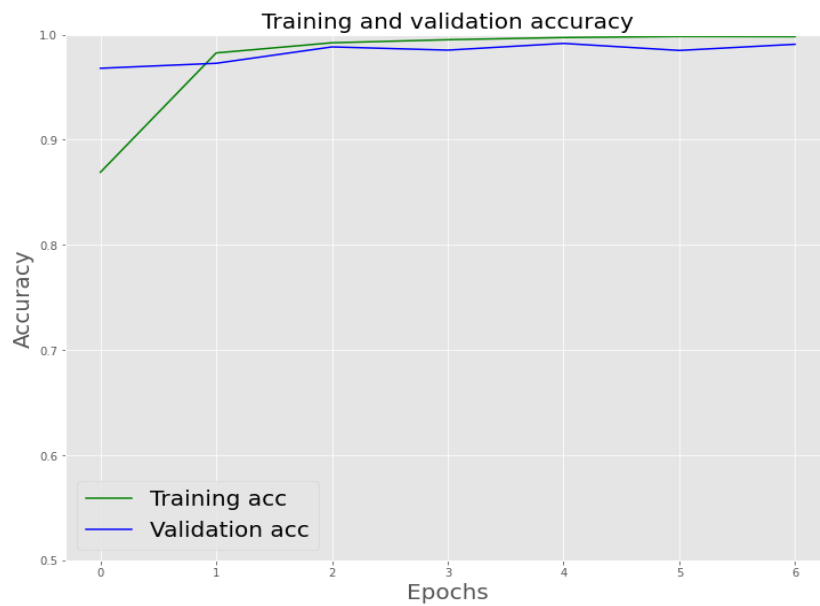
Figure: Classification Report

## RNN

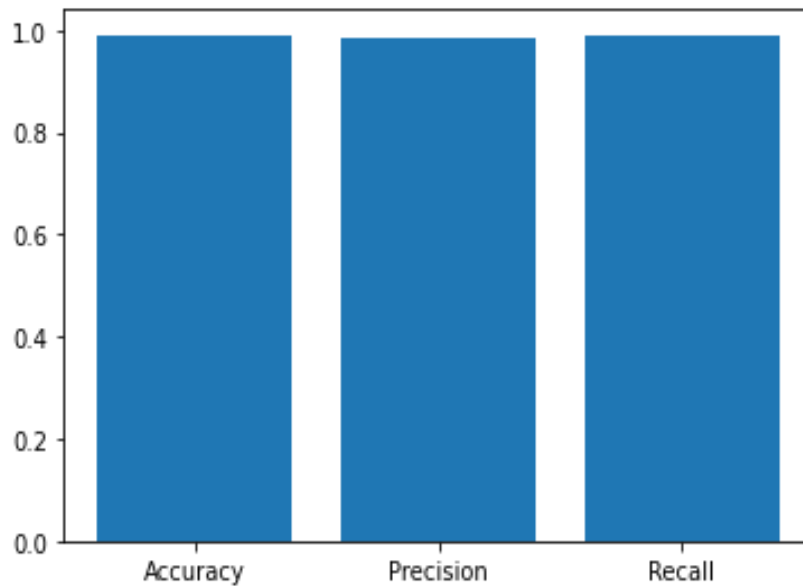
At the very beginning of our implementation, we have made a bar chart to see the distribution of both fake news and real news. This bar chart also helps us to identify the difference between these two types of news.



From the above figure, we can see that the proportion of fake news in our dataset is slightly greater than the proportion of real news. After this, the whole dataset went under pre-processing where certain columns were dropped, and certain columns were concatenated so that it becomes easier to process the dataset. When the whole dataset was done with pre-processing, the dataset was divided into two parts, one was the training data and the other was the testing data. The ratio of the testing data was kept at 20percent and the ratio of the training data was kept at 80percent. This model was trained for 10 epochs. It was found that the accuracy of every epoch kept on increasing. The accuracy started from 86.9percent and in the 3rd epoch, it reached 99.2percent. The accuracy went up-to 99.81percent which was the maximum. It was also observed that the loss was decreasing consistently, and it reached 0.82percent at the last epoch which shows that the model was very efficient. The following figure show the graph of accuracy and loss of the training and testing of data.

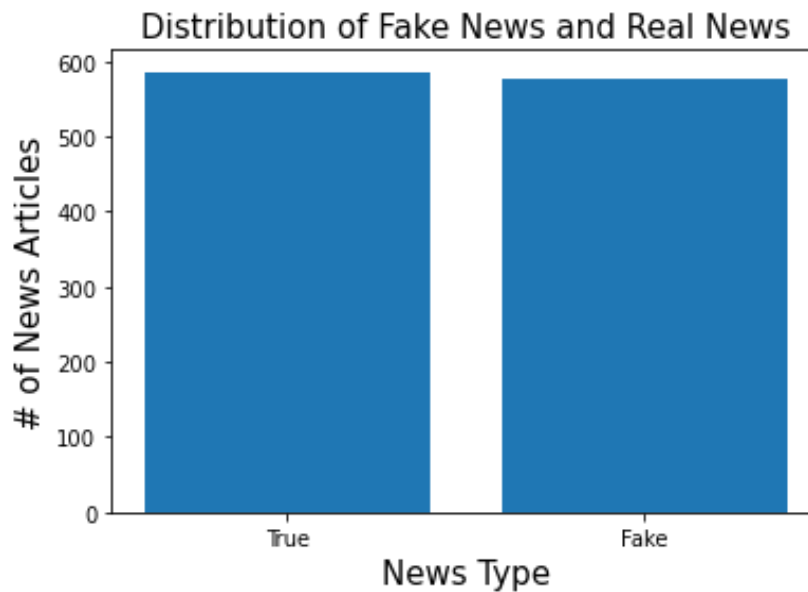


We have also calculated the accuracy, precision and recall value of our testing data set and the results showed that it was very efficient. The accuracy was 98.85percent, precision was 98.49percent and the recall value was 99.11percent.



### Logistic Regression

Initially overview of the dataset is represented by a graph which tells how much true and fake news the dataset contains. Bar chart is given below.



For this dataset fake and true news is almost 50/50. After the pre-processing of the given dataset, it went through the logistic regression model for fake news detection and showed result accuracy of about 92percent. Later accu-

racy, f1 score support etc. also were calculated and plotted with graph and has been shown. For simplification precision, recall and f1 score is shown in 100percent and support is as it was. For true news precision achieved was 92percent where for fake news the precision was 93percent. All the attributes are introduced in the bar chart given below.

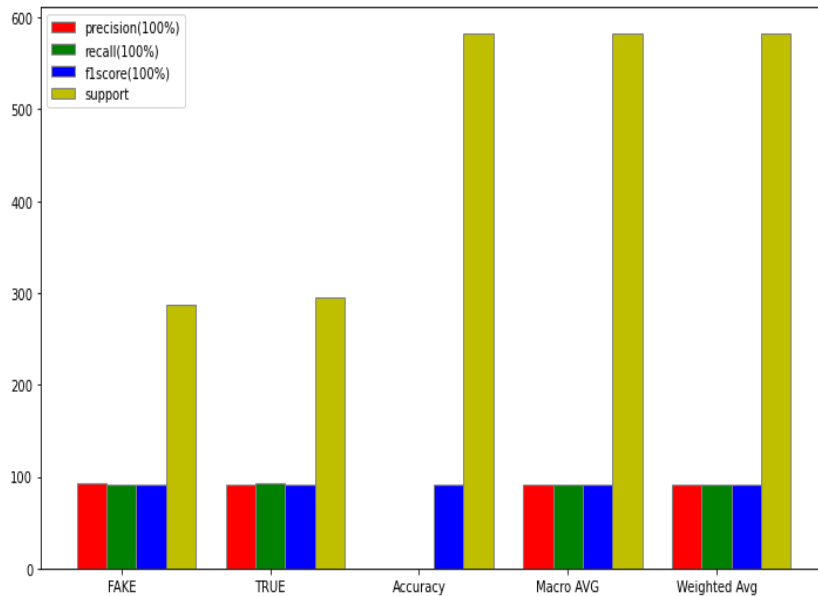


Figure: Classification Report

## 4. Conclusion

There were number of algorithms to use for fake news detection. Every algorithm works different and has various accuracy. In our project we couldn't apply all the algorithm rather we implemented SVM, Naïve Bayes, RNN and Logistic Regression. And for all the algorithm the accuracy and result were satisfactory. For different algorithm different dataset is used so it is hard to say which one worked the best.

### 4.1 Challenges

1. Finding same dataset for all the algorithms was hard. Only SVM and Naive Bayes had same dataset.
2. Different library of python was used and understanding them was sometimes complex.

3. Importing some library showed some error first and also version of different environment gave errors sometimes.

## **4.2 Limitations**

1. Our project only used dataset from Kaggle we didn't test for other data sources
2. A major flaw of the bag-of-words model is that it does not account for the ordering of words in a document, and as such any random permutation of words of a document would end up having the same representation. For example, the following sentences, although different semantically, would have the same representation under the BoW model:

The quick brown fox jumps over the lazy dog

The quick dog jumps over the lazy brown fox

3. Another issue with the BoW model is its inability to put adequate weights on important, representative words in the corpus vs frequently occurring words in the language like prepositions and pronouns. For example, using the BoW model, words like "the", "a", "an" which occur very frequently in English text would always end up with high counts.

## **4.3 Future directions**

1. All the algorithms can work on the same dataset.
2. Data preprocessing advancement can give more accuracy for the models.
3. CNN, DS-TREE can also be used for this project.

## References:

- [1] V. Agarwal, H. P. Sultana, S. Malhotra, and A. Sarkar, “Analysis of Classifiers for Fake News Detection,” 2019, doi: 10.1016/j.procs.2020.01.035.
- [2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media,” ACM SIGKDD Explor. Newsl., 2017, doi: 10.1145/3137597.3137600.
- [3] N. Ruchansky, S. Seo, and Y. Liu, “CSI: A hybrid deep model for fake news detection,” 2017, doi: 10.1145/3132847.3132877.
- [4] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, “Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter,” 2017, doi: 10.18653/v1/P17-2102.
- [5] “No Title.” <https://iq.opengenus.org/text-classification-naive-bayes/> (accessed Sep. 08, 2021).
- [6] “Java T Point.” <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm> (accessed Aug. 07, 2021).
- [7] “Towards Data Science.” <https://towardsdatascience.com/how-to-build-a-recurrent-neural-network-to-detect-fake-news-35953c19cf0b> (accessed Sep. 09, 2021).