CSE/ECE 343 : Machine Learning Project Report

Title : Fake News Detection

Satya Shambhavi Abhiroop Mareedu[#1] Shashank Chaurasia[*2] Jatin Yadav[#3] Bharat Soni[#4]

satya19204@iiitd.ac.in *shashank19105@iiitd.ac.in* *jatin19470@iiitd.ac.in* *bharat19461@iiitd.ac.in*

Abstract

Today Fake News has become a major problem wreaking havoc all over the world. True Information is crucial to Human life for important decision making. We form ideas about people or situations through access to Information. Due to Internet misinformation such as fake news, hoaxes, and violent opinions are widespread. Targeted delivery of content has put people in misinformation bubbles which are used to further one's agenda.This relates to politics, health and hate speech. This can impact us financially, health-wise, democratically and socially.Hence, we plan to propose a solution for fake news detection which incorporates sentiment as an important feature to improve accuracy.

1. Introduction

We propose Binary Classification models that we can use to classify if a statement is reliable or unreliable. This classification process has diverse practical applications as it automates the assignment of statements to either reliable or not with a very high accuracy. We classify based on the title, author, text, verbs etc.

2. Literature Survey

Fake news detection is a complicated and broad problem which has been approached in many different ways:

1. Fake News Detection Using Machine Learning Ensemble Methods [1] by Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf and Muhammad Ovais Ahmad used different ensemble machine learning techniques like Random Forests, bagging classifiers, boosting classifiers and voting classifiers using DTs, KNN, Logistic Regression, Linear SVM.

2. Fake News Detection Using Machine Learning Approaches [2] ,Z khanam, B N Alswel, H Sirafi and M Rashid used NLP NLTK libraries to clean the noise, performed feature extraction using lexical features such as word counts,adjectives,nouns.Also, applied TFIDF vectorizer.

3. Dataset Features

There are 20800 total data points in the first dataset.

- id: unique id for a news article (int)
- title: the title of a news article
- author: author of the news article
- text: the text of the article; could be incomplete
- label: a label that marks the article as potentially unreliable
    - 1: unreliable
    - 0: reliable

We notice that there are many null values in the title and author columns as compared to the text column.

```
id          0
title     558
author   1957
text       39
label       0
dtype: int64
```

Figure 1: Null values in dataset 1

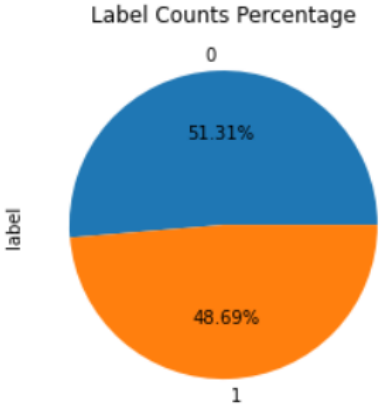

Figure 2: The proportion of labels, articles classified as fake (1) or not fake (0) in dataset 1

There are 4048 data points in the second dataset.

- URLs: link for article
- headline: the title of a news article
- body: author of the news article
- label: a label that marks the article as potentially unreliable
    - 1: unreliable
    - 0: reliable

```
URLs          0
Headline      0
Body         21
Label         0
dtype: int64
```

Figure 3: Null values in dataset 2

```
0    2137
1    1872
Name: Label, dtype: int64
```

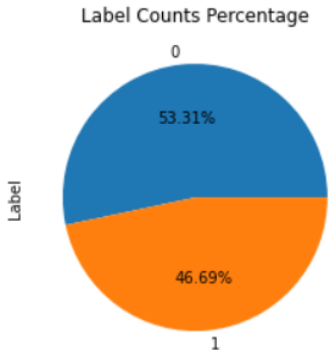

Figure 4: The proportion of labels, articles classified as fake (1) or not fake (0) in dataset 2

3.1 Data Preprocessing

3.1.1. Dataset 1

First we converted the text data into lowercase letters so it's easier to compute. Then we removed contractions from our text, meaning that words like “I’ll” were changed to “I will” so that the data is more uniform.

Then to avoid any more redundant data, we removed all characters from the data which were non ASCII and also those words which were single characters because they don’t contribute to the model training and take up more computational space.

All digits are also removed for the same reason. All punctuations are removed, characters like ‘ ; [ ] / . , - = which provide no fake/real information to the data were also removed.

The next step we performed in preprocessing was that we removed all stop words, small words for the purposes of saving space and time in processing of large data. Words like “as” “in” “on” and stemmed the data using PorterStemmer.

The null values in the title column were set to False and the rest were set to True. The null values in the author column were set to ‘unknown’ and the rest remained the same.

3.1.2. Dataset 2

The same preprocessing was done for dataset 2. The null values were replaced with blanks in ‘body’ column. The headline and body column were combined and our models were trained on them.

4. Methodology

We applied binary classification on a total of 20,800 and 4048 statements. We performed a 80:20 train-test split using TfidfVectorizer for the train set which calculates TFIDF (Term Frequency Inverse Document Frequency) using the following formula for the text data.

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Figure 3: Calculation of TFIDF

Uncommon words have higher TFIDF and common words have lower or almost 0 TFIDF.

We only chose the author name and title to train our model. To see how our model performs compared to using text body and title, we have used the second dataset and performed the same model training on its title+body and analysed the results of both.

4.1 Classification

The following classification models were and will be used Logistic Regression, Naive Bayes, Decision Trees, Random Forests, KNN to optimize various parameters in the mentioned models. GridSearchCV and 10-fold cross validation.

4.1.1. Logistic Regression

We performed Logistic Regression on the processed data and got the following result on the test data:

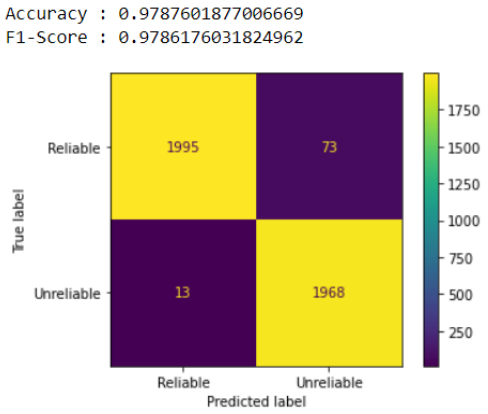

Figure 4: Result of Logistic Regression model (Dataset 1)

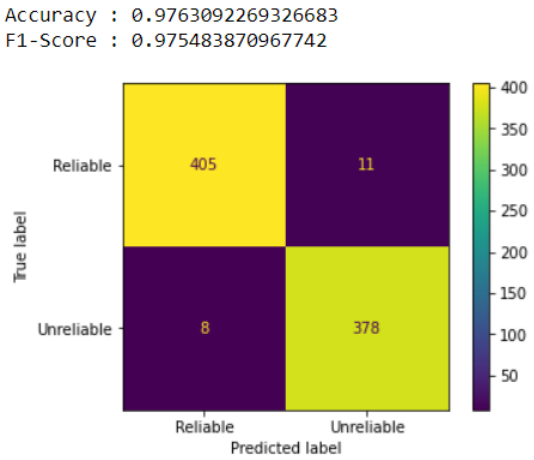

Figure 5: Result of Logistic Regression model (Dataset 2)

Logistic regression is a supervised learning classifier used to predict the probability of a target variable. The nature of the target variable is dichotomous, which means there would be only two possible classes.

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms which is used for various classification problems.

4.1.2. Naive Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among the attributes. It uses the probabilities of label given a sample to classify data samples. There are three different types of Naive Bayes algorithms that we will be using:

4.1.2.1. Gaussian Naive Bayes

We performed Gaussian Naive Bayes on the processed data and got the following result on the test data:

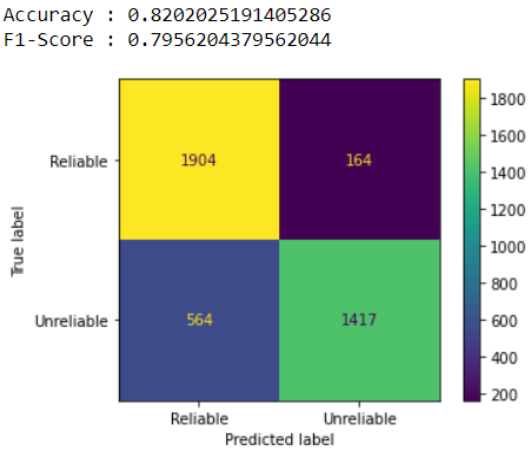

Figure 6: Result of Gaussian Naive Bayes (Dataset 1)

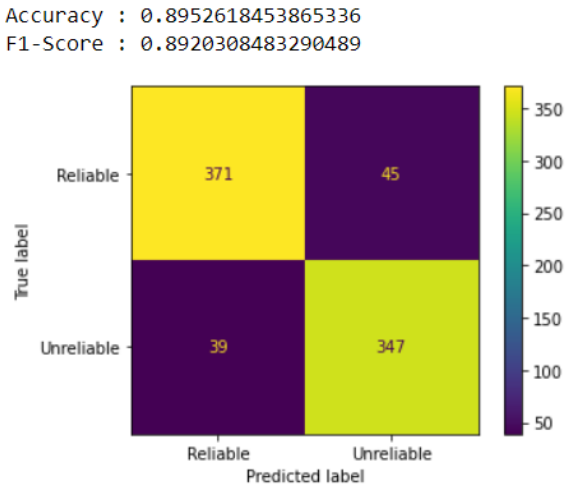

Figure 7: Result of Gaussian Naive Bayes (Dataset 2)

It is the simplest Naïve Bayes classifier having the assumption that the data from each label is drawn from a simple Gaussian distribution when the data is continuous and not discrete.

4.1.2.2. Multinomial Naive Bayes

We performed Multinomial Naive Bayes on the processed data and got the following result on the test data:

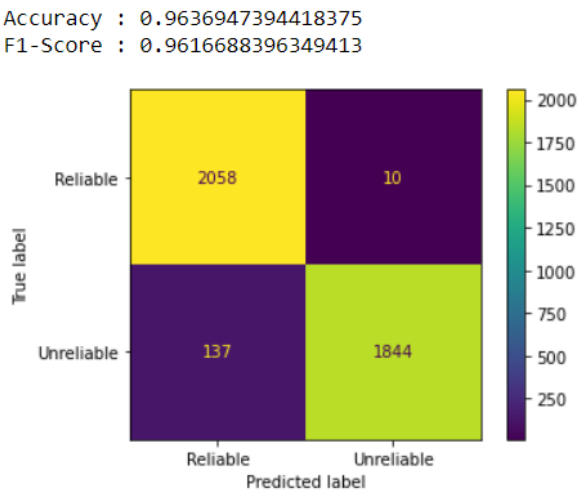

Figure 8: Result of Multinomial Naive Bayes (Dataset 1)

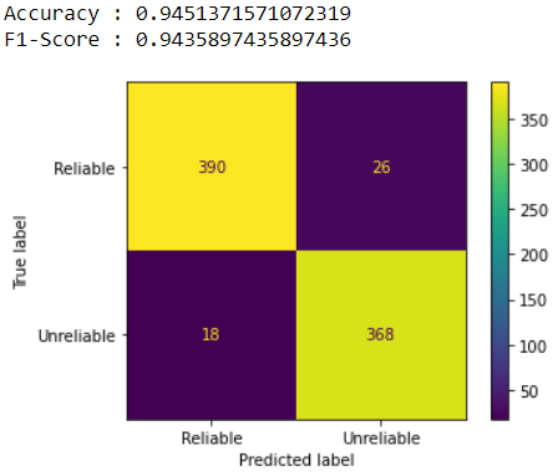

Figure 9: Result of Multinomial Naive Bayes (Dataset 2)

Another useful Naïve Bayes classifier is Multinomial Naïve Bayes in which the features are assumed to be drawn from a simple Multinomial distribution. Such kind of Naïve Bayes are most appropriate for the features that represents discrete counts.

4.1.2.3. Bernoulli Naive Bayes

We performed Multinomial Naive Bayes on the processed data and got the following result on the test data:

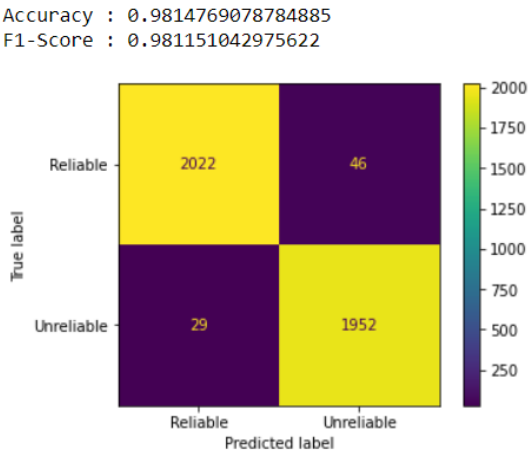

Figure 10: Result of Bernoulli Naive Bayes (Dataset 1)

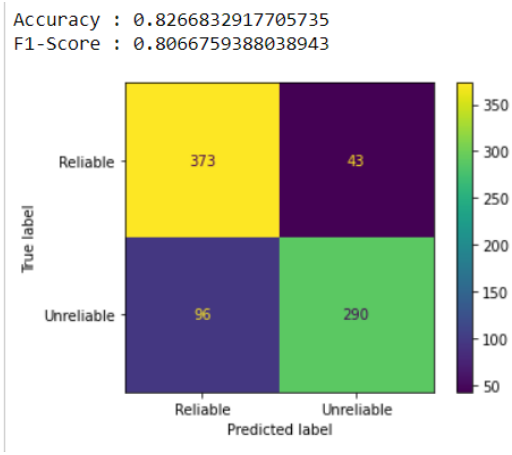

Figure 11: Result of Bernoulli Naive Bayes (Dataset 2)

In Bernoulli Naïve Bayes, features are assumed to be binary (0s and 1s). Text classification with 'bag of words' model can be an application of Bernoulli Naïve Bayes. Hence Bernoulli is best for text classification.

4.1.3. Decision Trees

By learning simple decision rules inferred from past data, the purpose of employing a Decision Tree is to develop a training model that can be used to predict the class or value of the target variable (training data)

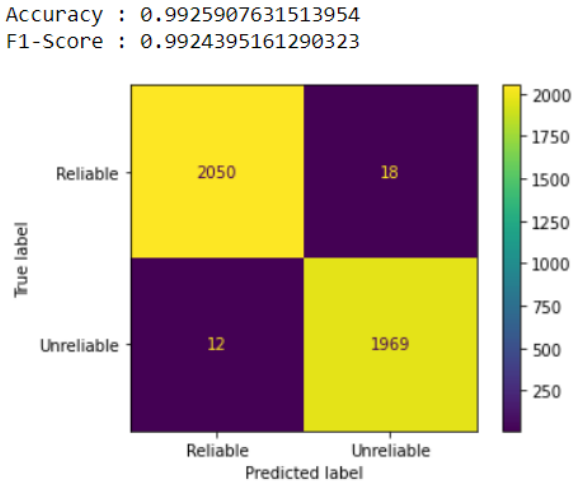

Figure 12: Result of Decision Trees (entropy) (Dataset 1)

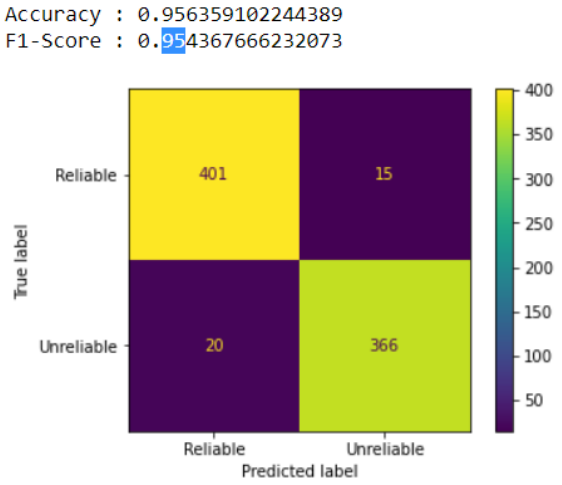

Figure 13: Result of Decision Trees (entropy) (Dataset 2)

Entropy gave better result than gini split in decision tree making.

4.1.4. Random Forests

Random Forests are based on the idea of constructing a large number of decision trees, each of which produces a unique outcome. The random forest takes up the results, which are anticipated by a huge number of decision trees. The random forest selects a subcategory of attributes from each group at random to ensure that the decision trees are varied.

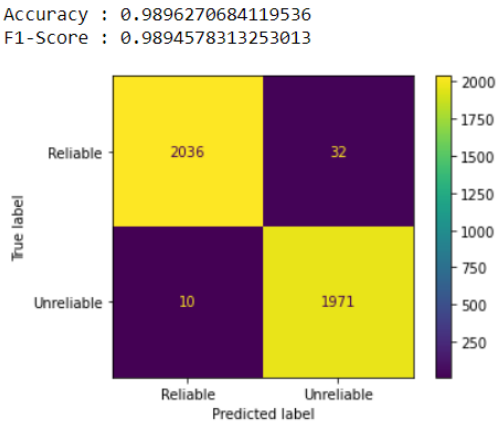

Figure 14: Result of Random Forest (entropy) (Dataset 1)

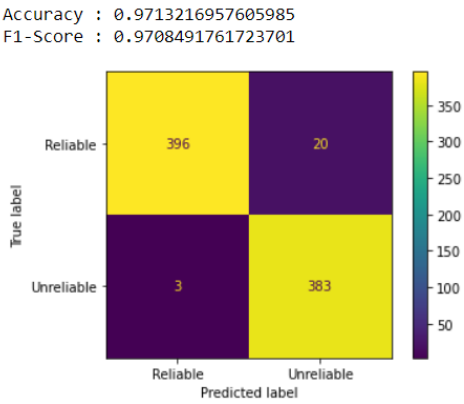

Figure 15: Result of Random Forest (entropy) (Dataset 2)

4.1.5. K Nearest Neighbours

KNN works by calculating the distances between a query and all of the instances in the data, picking the K closest examples to the query, and then voting for the most frequent label (in the case of classification) or averaging the labels (in the case of regression).

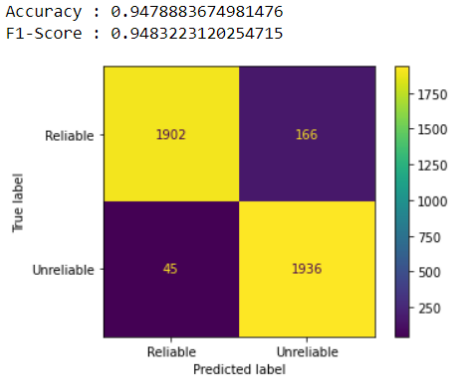

Figure 16: Result of KNN (Dataset 1)

```
Accuracy : 0.9089775561097256
F1-Score : 0.903054448871182
```

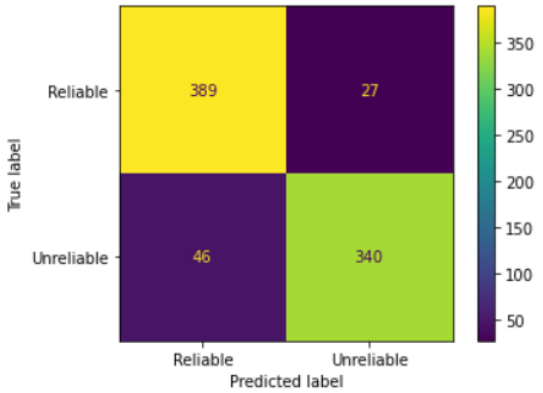

Figure 17: Result of KNN (Dataset 2)

4.1.6. Support Vector Machine

SVM produces high accuracy with less computational power. It finds a hyperplane in n dimensional space where n is the number of features. For binary classification it finds the hyperplane which has maximum distance from the data points of both classes.

```
Accuracy : 0.9920968140281551
F1-Score : 0.9919436052366566
```

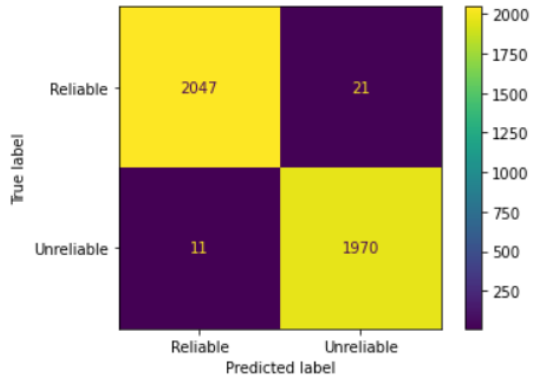

Figure 18: Result of SVM (Dataset 1)

```
Accuracy : 0.9875311720698254
F1-Score : 0.9870801033591732
```

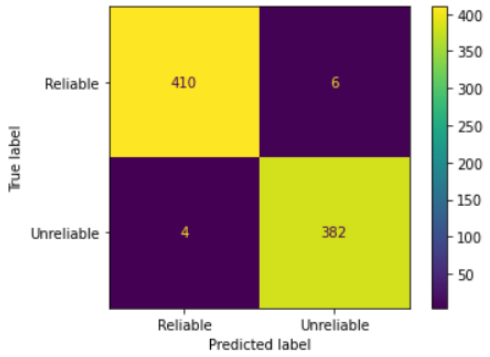

Figure 19: Result of SVM (dataset 2)

5. Result Analysis:

ML Model	Accuracy for Dataset 1	Accuracy for Dataset 2
Logistic Regression	97.87 %	97.63 %
BernoulliNaiveBayes	98.14 %	82.66 %
Gaussian Naïve Bayes	82.02 %	89.52 %
Multinomial Naïve Bayes	96.36 %	94.51 %
Decision Trees	99.24 %	95.63 %
Random Forests	98.96 %	97.13 %
K Nearest Neighbours	94.78 %	90.89 %
Support Vector Machine	99.20 %	98.75 %

Figure 20: Overall Result Table for all models

Overall the models trained better on dataset 1. This is mostly because of the nature of the data. We had trained the data on the Title and author of the article. However in dataset 2, we had only used the Headline + body of the articles. Since we couldn't extract the author names into different columns due to data unavailability. Let us analyse the results.

5.1. BernoulliNB:

Since the data is assumed to be either 0s or 1s, Bernoulli Naive Bayes is usually used for classification with the 'bag of words' model. Hence Bernoulli is best for text classification and we got the highest accuracy with this model even over logistic regression.

$$f(x) = \begin{cases} p^x * (1-p)^{1-x} & if\ x = 0,1 \\ 0 & otherwise \end{cases} = \begin{cases} p & if\ x = 1 \\ 1-p & if\ x = 0 \end{cases}$$

Figure 20: Bernoulli Naive Bayes

However in case of dataset 2, Bernoulli performs the worst since we only had 1 feature for classification. This is bad since bernoulli assumes that data is either 0 or 1.

5.2. Logistic Regression

Logistic regression is intended for binary (two-class) classification problems. It will predict the probability of an instance belonging to the default class, which can be snapped into a 0 or 1 classification in this project, it's either fake or real news.

It is less inclined to over-fitting but it can overfit in high dimensional datasets. Since our dataset is not high dimensional. It works great in both cases.

5.3. Decision Tree

On the one hand, the gini criteria is significantly faster due to its lower computing cost. The results achieved utilizing the entropy criterion, on the other hand, are marginally better. The Gini Index has values between 0 and 0.5, whereas the Entropy has values between 0 and 1.

It gives the highest accuracy comparable to SVM in dataset 1 and acceptable accuracy in case of dataset 2. Since data wasn't high dimensional, performing splits in the second dataset may have proved to be more challenging and we can say that is the reason for its poorer performance compared to dataset 1.

5.4. Support Vector Machine

Comparing the results, we see that SVM performed the best in both cases and we would recommend this model to be used for future fake news classifications. SVM calculates for the hyperplane between the data points of both classes. Since our data was to be binary classified, SVM proved to be the best model for this job.

5.5. Gaussian Naive Bayes

This model performed very poorly in both cases. This is because Gaussian Naive Bayes can be useful to be applied on real or continuous-valued datasets rather than categorical or discrete-valued features. Since we don't have many dimensions to our data and it is not continuous valued data, Gaussian performed poorly.

6. Conclusion

6.1 Outcomes

We are pleased to inform you of our hypothesis that the titles play a significant role in fake news detection.
The Metric scores obtained by our classification models were very promising and posed a similar score with respect to previous work done on this topic. The models also worked promisingly across both the data sets with a very high accuracy. The best accuracy for dataset 1 is Decision tree with an accuracy of of 99.25% and SVM with an accuracy of 98.75% for dataset 2 and it was found that SVM performed best for both datasets as well.

The reason for performing on two datasets was to compare the training of models based on (Author + Title) vs (Title + Body) of news articles. Similar results were achieved on both. However it was concluded that (Author + Title) gave better accuracy for less computational time. Hence it can be concluded that fake articles arise from clickbait titles and the same authors are more likely to spread fake news. The content of a new article is not as useful as the other two when training binary classification.

6.2 Learning from the project

- Team Work
- Handling of Large Data
- Detection of Outliers
- Handling of null/NaN values
- Choosing models depending on evaluation metrics for supervised learning based on binary classification.
- How to use NLP tools to turn data into usable form for ML classification

6.3. Future Work

We could use these models to train fake news classification on different languages like Hindi, French, Italian, etc. Chinese is the most popular language in the whole world .Using appropriate NLP tools, we suggest that ML models for Chinese fake news classification can be done.

We also suggest that fake tweet classification can be done for tweets on twitter, since the world is evolving and social media is one of the major information sources for the newer generations.

6.4 Member Contribution

Bharat Soni: Dataset Preprocessing, Removing Stop-Words, Logistic Regression, Naive Bayes, TFIDF, Study and Research, SVM,KNN, Analysis ,Report, Second Dataset preprocessing and training

Shashank Chaurasia: Dataset Preprocessing,Literature Review, TFIDF, Study, Logistic Regression and Research, SVM, Decision Trees, Result Analysis, Second Dataset

preprocessing and training,SVM, KNN, Analysis ,Report

Jatin Yadav: Feature Extraction, metrics comparison, Study and Research, Analysis, Naive Bayes,Decision Tree,Random Forest, Analysis ,Report

Satya Shambhavi Abhiroop Mareedu: Feature extraction, Literature Review, Analysis, Logistic Regression Study and Research, Decision Tree,Random Forest., Analysis ,Report

We have made sure equal work was done by all.

7. References

1. Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, Muhammad Ovais Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods", Complexity, vol. 2020, Article ID 8885861, 11 pages, 2020. https://doi.org/10.1155/2020/8885861

2. Z Khanam et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1099 012040. https://iopscience.iop.org/article/10.1088/1757-899X/1099/1/012040