FAKE NEWS DETECTION USING NLP

By

Anusha Ayyagari

Harikrishna Bhumani

Department of Data Science

University of Maryland Baltimore County

DATA 606: Capstone Project

Dr. Professor Barber

December 13 2022

ACKNOWLEDGEMENT

Firstly, we would like to thank our professor Dr. Professor Barber for the immense support and valuable advice for the completion of the project.

We also would like to thank our program director Dr. Ergun Simsek for providing the valuable curriculum and the courses that helps us to complete the project

TABLE OF CONTENTS

Page

Title Page i

Acknowledgement ii

Table of Contents iii

List of Figures and Tables iv

List of Abbreviations v

Abstract vi

Purpose vii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 LITERATURE REVIEW 3

CHAPTER 3 DATASET 5

CHAPTER 4 METHODOLOGY 7

CHAPTER 5 DATA ANALYSIS 9

5.1 Visualizations 11

5.2 Statistical Analysis 17

CHAPTER 6 MODEL DESIGN 20

6.1 Model Development 20

6.1.1 Naive Bayes 20

6.1.2 SVM 20

6.1.3 Logistic Regression 21

6.1.4 XGBoost 21

6.1.5 Bi-Directional LSTM 21

6.1.6 GRU 22

6.2 Experiment Results 23

6.3 Discussion 25

CHAPTER 7 LIMITATIONS 26

CHAPTER 8 FUTURE SCOPE 27

CONCLUSION 28

REFERENCES 29

LIST OF FIGURES AND TABLES

Figure **Page**

4.1 Proposed Model Architecture…… 7

5.1 Distribution of labels in the dataset……11

5.2 Distribution of sentiment in the dataset…… 12

5.3 Distribution of sentiment for each label…….12

5.4 Top 10 originators of information in the dataset……13

5.5 Top 10 originators of true information in the dataset…….13

5.6 Top 10 originators of misinformation in the dataset……...14

5.7 Top 10 platforms in the dataset…….. 14

5.8 Top 10 platforms of misinformation in the dataset…….15

5.9 Top 10 platforms of true information in the dataset……15

5.10 Distribution of amount of misinformation through the years……..16

5.11 Distribution of the amount of true information through the years……...16

5.12 Top 10 fact checkers in the dataset…….. 17

6.2.1 ROC curves for Naive Bayes Classifier ……. 24

6.2.2 ROC curve for SVM…… 24

6.2.3 ROC curve for Logistic Regression…… 24

6.2.4 ROC curve for XGBoost Classifier……. 24

Table **Page**

3.1 Dataset Information……... 6

4.1 Data frame after cleaning the data……. 9

6.1 Model results……. 23

LIST OF ABBREVIATIONS

NLP	-	Natural Language Processing
BOW	-	Bag of Words
NLTK	-	Natural Language Processing Tool Kit
SVM	-	Support vector machine
XGBoost	-	Extreme Gradient Boosting
LSTM	-	Long Short Term Memory
RNN	-	Recurrent Neural Network
ROC	-	Receiver Operating Characteristic
VADER	-	Valence Aware Dictionary and Sentiment Reasoner

ABSTRACT

Spreading of false news and hoaxes has increased exponentially over the last decade due to increased internet usage and the advent of social media platforms. Detecting these false statements and taking appropriate action has now become a necessity as it leads to catastrophic consequences. Currently many social media platforms rely on content moderators and user's feedback for detecting fake news which takes up a lot of time and manual effort. This research aims to automate the task of identification of false information on crime by using Machine learning algorithms so that these posts can either be taken down immediately or flagged for further investigation by the platforms where they were posted. The research consists of three phases: data collection and analysis phase, feature extraction and transformation phase and prediction phase. In the data collection process, labeled data is collected from the website PolitiFact, after which various transformations are applied to clean the data and gain some insights about the dataset. In the next phase, various natural language processing methodologies are applied to transform the text data into data suitable for machine learning models to process. In the final model building and detection phase various supervised machine learning algorithms as well as a few deep learning algorithms are used to classify if the given statement is true information or not. After comparing all the models on the test set it was observed that the gradient boost classifier was the best model outperforming even the advanced deep learning models due to which it was selected as the final model.

PURPOSE

The purpose of our research is to create a model that will detect if a given piece of information on crime is factual or fake based on its features like words and phrases. To achieve this, we apply supervised machine learning algorithms on a labeled dataset, that was manually fact checked and classified by independent journalists and was scraped from a reliable website. Then, the BOW model was used to convert the text into a feature vector by counting the occurrence of words in a document. We tested different classification algorithms on the unseen data and the model that performed the best can be used in future to either delete the false posts automatically or flag the posts so that the users are informed about the nature of the information they have posted.

Our research tries to answer the below questions:

1. Can we determine if the given information about crime is factual or fake.
2. Are the platforms where the statements were made and the labels dependent ?

   H0(Null Hypothesis): The Label and platform where the information was posted are independent.

   H1(Alternate Hypothesis): The Label and platform where the information was posted are dependent.

3. Are the sources of information and labels dependent?

   H0(Null Hypothesis): The Label and sources are independent.

   H1(Alternate Hypothesis): The Label and sources are dependent.

4. Are the negative sentiment and the label dependent?

   H0(Null Hypothesis): The Label and negative sentiment are independent.

H1(Alternate Hypothesis): The Label and negative sentiment are dependent.

5. What is the average time required to check if the statement is true or false manually.
6. Does machine learning algorithms perform better than deep learning algorithms for predicting the class of the statement

CHAPTER 1

INTRODUCTION

Fake news can be defined as a set of elaborate lies or hoaxes that are deliberately fabricated to mislead or defame an individual, a community or an organization. With the widespread reliance on social media for information and unchecked growth of such platforms, false news reaches millions of people in a matter of few seconds which can lead to devastating consequences for the society [1]. In recent years, reluctance of social media executives to proactively identify and censor false information on their platforms to promote free speech has led to several civic unrests like the Rohingya genocide, healthcare nightmares like unverified COVID treatments and threats to national security. Additionally, an increase in the circulation of False information has also seen an increase in Cyberbullying, public shaming, mental health crisis and in extreme cases suicide and wrongful convictions. Given this dangerous nature of false information it is necessary to identify this information and remove it from social media platforms as soon as possible [2].

Social media companies like Facebook and YouTube have dedicated content monitoring teams that continuously monitor the content posted on the platforms and take appropriate actions. However, as these platforms publish millions of posts in a day it is impossible to screen all of them manually, hence it is critical to develop a system that can automatically detect false information immediately after it is posted. One of the most effective solutions for fake news detection is to identify the original source of the social media post, by understanding the intention of the source, the system can decide if the information is true or false [3]. Despite the relative effectiveness of such models the problem arises when the same source, whether a human or a machine, might produce both truthful and incorrect or misleading information, which makes it difficult to accurately determine the nature of the information [4].

There are multiple solutions used for identification of fake news. The approach taken in this research is first to build a corpus from a reliable source, then clean the data and make use of different libraries and techniques to perform data preprocessing to gain useful insights and

statistics form the dataset and then perform pretext operation such as tokenization and handling stop-words to remove the noise in the data. Then extract the features and use different supervised models to achieve the improved accuracy in predicting if the information is true or false. This research mainly focuses on identifying the false information in context of crime and uses sentiment analysis to understand the influence of sentiment in determining if the information is true or false.

CHAPTER 2

LITERATURE REVIEW

Despite fake news detection in social media getting attention recently, there has been a flux of research and publications on the issue. There are a variety of models to predict fake news which range from traditional machine learning models to complex deep learning models and, in recent times, Transformer models as well. The public availability of benchmark datasets like LIAR [5], FakeNewsNet [6]and Fake News dataset on Kaggle [7] has piqued the interest of many machine learning enthusiasts to conduct their own research on the topic and share their insights.

The authors [8] summarized the task of identifying false messages as being done in three stages of processing, feature extraction and classification. Their proposed model is a hybrid classification model as it is a combination of KNN and random forest. The execution of the proposed model is analyzed for accuracy and recall. The final results improved by up to 8% using a mixed false message detection model. Support Vector Machine and Naive Bayes Classifier are frequently used classification models, the model proposed by [9] used naïve Bayes classifier to detect fake news. This method was trained and tested with records from various sources like Facebook and had an accuracy of 74%. The authors claim that neglecting punctuation errors resulted in poor accuracy of the model.

The authors [10] used the classic and state-of-the-art classifiers, including k-Nearest Neighbors, Naive Bayes, Random Forests, Support Vector Machine with RBF kernel, and XGBoost for detecting false news on a dataset that consists of 2282 BuzzFeed news articles related to the 2016 U.S. election. They evaluated these models on Area Under the Curve and F1 score and concluded that the SVM and XGB models performed better than the rest of the models. The authors [11] experimented with Decision Tree,k-Nearest Neighbors, Naive Bayes, Random Forests, Support Vector Machine and XGBoost on the LIAR dataset to detect fake news. The highest accuracy of 75% was obtained by the XGBoost classifier in their research.

When it comes to Neural Networks based false news detection models, variations of Recurrent Neural Network (RNN) are very popular choice, especially Long Short-Term Memory (LSTM), which solves the vanishing gradient problem so that it can capture longer-term dependencies.

[12] proposed DeClarE, an end-to-end neural network model, employs evidence and counter-evidence extracted from the web to support or refute a claim. The authors achieved an overall 80% classification accuracy on four different datasets, by training a bi-directional LSTM model with attention and source embeddings.

Convolutional neural networks (CNN) are also widely used since they succeed in many text classification tasks. [13] have proposed a deep convolutional neural network model that does not rely on extracting hand-crafted features. Instead, the model is designed to learn the discriminatory features through the deep neural network automatically. They had created a deep convolutional neural network (CNN), and a set of features that are extracted at each layer. The model achieves an accuracy of shown outstanding performance 98.36% on large-scale real-world fake news datasets. [14] proposes a hybrid model framework that had a 1D CNN immediately after the word embedding layer of the LSTM model and achieved an accuracy of 80% in fake tweets prediction. Attention mechanisms are often incorporated into neural networks to achieve better performance. [15] used an attention model that incorporates the speaker's name and the statement's topic to attend to features first and then feed the weighted vectors into an LSTM. Doing this increased the accuracy of the model by about 3%.

CHAPTER 3

DATASET

PolitiFact is a Pulitzer Prize winning political fact-checking website. It is owned by the Poynter Institute, a nonprofit school for journalists. In their own words Politifact claims to be "a nonpartisan fact-checking website to sort out the truth in American politics" [5]. The fact-checkers manually examine the posts from various reliable sources and provide feedback to the social media companies as to the accuracy of the claims made in the posts. The high ethical standards and the credibility of the journalists working for the website, makes this a trusted source for constructing fake news dataset [5]. Benchmark datasets like LIAR dataset [6] and FakeNewsNet [7] were built from PolitiFact making this the best choice for this research.

In this research we are mainly focused on detecting false information about crime, hence we extract the data only from the section's crime, guns, and marijuana of the website. The website classifies the information into six fine-grained labels for the truthfulness ratings: pants-fire, false, barely true, half-true, mostly true, and true. But for the convenience of our research, we later reduce the labels to only true and false. The data is scraped from the website using Beautiful Soup Library in python and initially is in the below format.

Name	**Description**	**Data Type**
Statement	Post made by a person	text
Link	Html page to access the post	text
Date	Fact checker and the date when the fact checking was completed	text
Source	Originator of the information	text
context	Platform where the information originated and origin date	text

MsgContent	Article or information that we are trying to classify	text
Label	Type of information	text

Table 3.1: Dataset Information

CHAPTER 4

METHODOLOGY

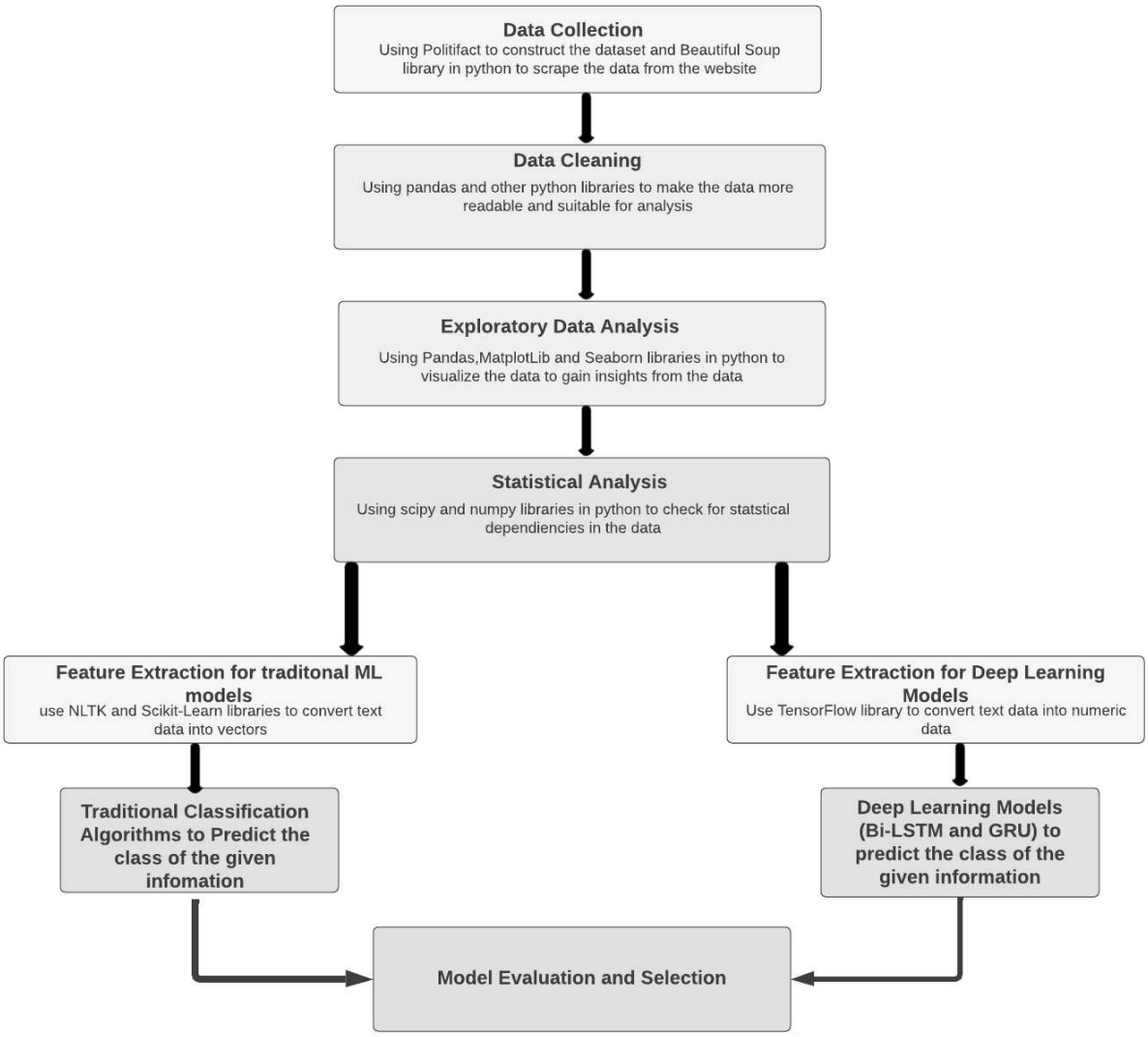

Figure 4.1: Proposed Model Architecture

After collecting the data, we start cleaning the data so that we can derive insights from it and then convert it into a format suitable for classification algorithms. First step in this process is to split the ‘Date’ column into columns ‘fact checker’ and ‘date’. Where fact checker provides information about the person who has verified the data and assigned the appropriate label, and ‘date’ column represents the date when fact checking was completed. Then the column ‘Context'

is split into columns ‘date stated’ and 'stated platform’, where ‘date stated’ provides information about the date when the statement was made, and the 'stated platform’ provides information about the media platform where the statement was made. After splitting the columns 'Date' and 'context' are dropped as they are no longer needed.

Further cleaning is performed in columns 'fact checker' and 'date stated' by removing unnecessary information and making the columns more readable. We wrote a ‘cleaner’ function to remove the alphanumeric characters in the text and make use of the NLTK library to tokenize the text and remove any words that are not defined in the NLTK corpus word module. The function is then applied to columns ‘MsgContent’ and ‘Stated source’.

We then proceed to drop rows with labels as ‘full-flop’,’ no-flip’ and ‘half-flip’ as they provide no insight for this research. Then the columns ‘date stated’ and ‘date’ are converted to datetime datatype and miscellaneous data cleaning is performed. Additionally, ‘Stated platform’ column is modified to reflect generalized media platforms so that it can be used for further data analysis.

Post cleaning the data, we start exploring the dataset and try to gain insights from it. We start by creating a new column called ‘days difference’ which provides information about the number of days required by the journalists to manually fact check the information and assign the appropriate label to it. We took the difference between the columns ‘date verified’ and ‘date stated’ to obtain this value. We are using the pre-trained sentiment analysis library called VADER in python to identify the sentiment of the text in the ‘MsgContent’ column and assign the appropriate polarity score. The SentimentIntensityAnalyzer function in the VADER library assigns something called the compound score to the text, this metric calculates the sum of all the lexicon ratings which have been normalized between -1 (most extreme negative) and +1 (most extreme positive). In our model, we assume that a compound score less than 0 is a "negative" sentiment, a score greater than 0 is “positive” and a score of 0 is “neutral”. The final data set is in the below format

Name	**Description**	**Data Type**

Statement	Post made by a person	text
Link	Html page to access the post	text
Source	Originator of the information	text
body	text of the tweet	text
MsgContent	Article or information that we are trying to classify	text
Label	Type of information true or false	text
fact_checker	Person who assigned the initial label	text
date_verified	Date when the fact checking about the information was completed	Date
date_stated	Date when the article was posted	Date
stated_platform	Media platform where the article was posted	text
days_difference	Amount of time required to fact check the information	numeric
sentiment	Sentiment of the article	text
compound	Polarity score of the sentiment	numeric

Table 4.1: Data frame after cleaning the data

To convert the problem from multi class classification to binary classification, we encode the labels ‘barely-true’, ’pants-fire’ and 'false’ as 0, because it can be inferred that the information associated with these labels is false and misleading. For the labels ‘true’, ’mostly true’ and ‘half-true’ we encode them as 1, because the information associated with these labels is either true or have some factual foundation. We then create data frames ‘time_series_data’ to check the distribution of the amount of information over the years and ‘sentiment_data’ to check the distribution of each sentiment per label. We use inbuilt python libraries like Matplotlib and Seaborn to visualize the data and libraries like NumPy and SciPy to perform statistical analysis. The results of the Exploratory Data Analysis and Statistical analysis will be discussed in upcoming sections.

To make the data fit for machine learning models we convert the text data in the ‘MsgContent’ column into numeric data. We first start by cleaning the text and removing any alphanumeric characters and then split the data into training and test datasets. Using the CountVectorizer function we transform a given text into a vector based on the frequency (count) of each word that occurs in the entire text. Then the TfidfTransformer function is used to transform text into a meaningful representation of numbers which is used to fit a machine algorithm for prediction. For the deep learning models, we use the Tokenizer function from TensorFlow to split the texts into tokens(words) while keeping only the most occurring words in the text corpus. The num_words parameter in the function keeps a prespecified number (in this case 5000 words) of words in the text only. As the deep learning models expect that each sequence (each training example) will be of the same length (same number of words/tokens) we pad all the sentences using pad_sequences functions from TensorFlow. Additionally, we convert the labels to categorical values (one hot vector encoding). Once the preprocessing is completed, we use this data to train and test the machine learning and deep learning models. After evaluating all the models, we select the best performing model and save the model.

CHAPTER 5

DATA ANALYSIS

In this section we will discuss the evaluation for our model. We will first start by discussing the results of exploratory data analysis and statistical analysis.

5.1 Visualizations:

1. Visualizing the distribution of labels in the dataset.

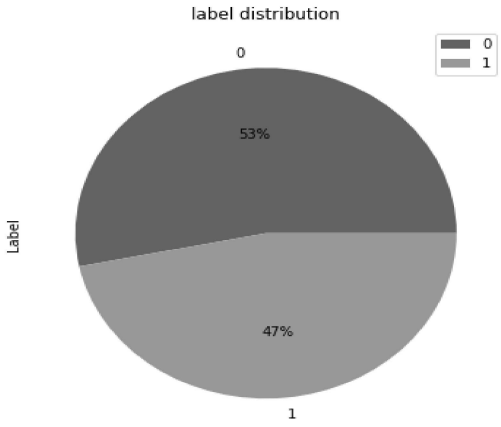

Figure 5.1: Distribution of labels in the dataset

The false label makes up 53% of the dataset and the true label makes 47%. We can conclude that this is a balanced dataset.

2. Visualizing the distribution of sentiment over the dataset

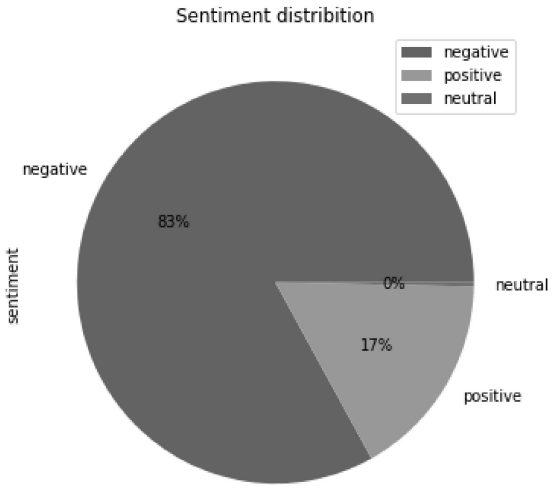

Figure 5.2: Distribution of sentiment in the dataset

The negative sentiment is the most dominant sentiment in our dataset with 83% of the articles having negative sentiment in them and only 17% of them having any positive sentiment.

3. Visualizing the distribution of different sentiments for each label:

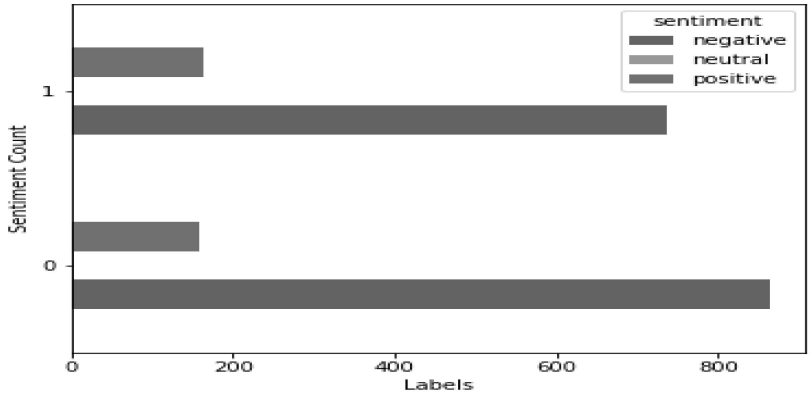

Figure 5.3: Distribution of sentiment for each label

For both the True and False label the number of articles with negative sentiment outnumber the articles with positive sentiment. We analyze this further to understand how the negative sentiment impacts the label of the article.

4. Visualizing the top 10 sources (owners of the post) of data.

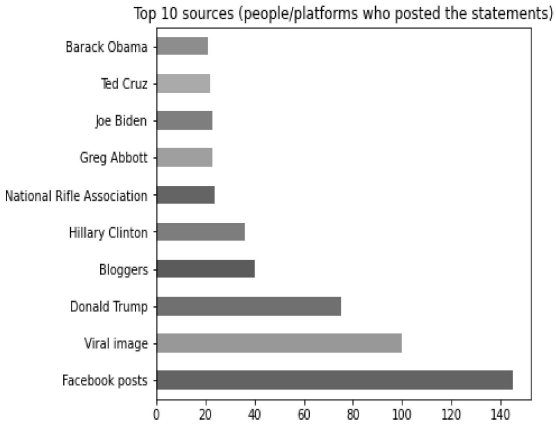

Figure 5.4: Top 10 originators of information in the dataset

Anonymous Facebook is the major source for building our dataset followed by viral images and Donald Trump.

5. Visualizing the top 10 sources (owners of the post) who posted true information.

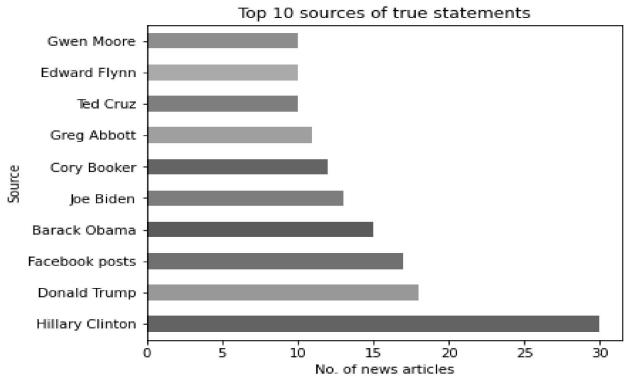

Figure 5.5: Top 10 originators of true information in the dataset

As per our dataset, Hilary Clinton made the greatest number of true statements followed by Donald Trump and anonymous Facebook posts.

6. Visualizing the top 10 sources (owners of the post) who posted misinformation.

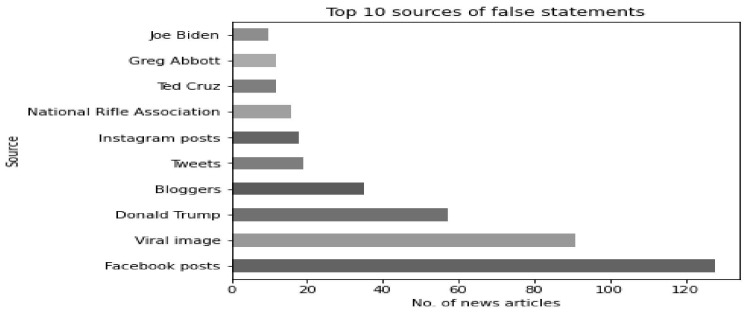

Figure 4.6: Top 10 originators of misinformation in the dataset

As per our dataset, anonymous Facebook posts made the greatest number of false statements followed by Viral images and Donald Trump.

7. Visualizing the top 10 platforms of data

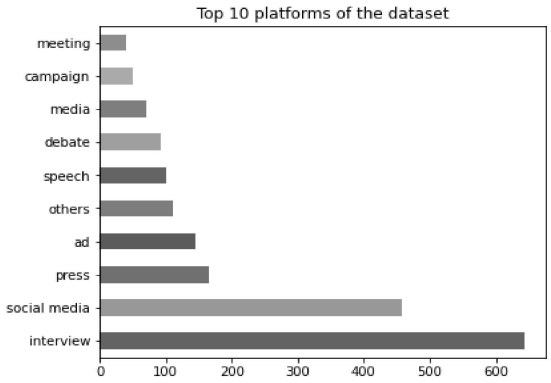

Figure 4.7: Top 10 platforms in the dataset

As per our dataset, interviews was the popular media platform where the statements were made followed by social media and press.

8. Visualizing the top 10 platforms for misinformation.

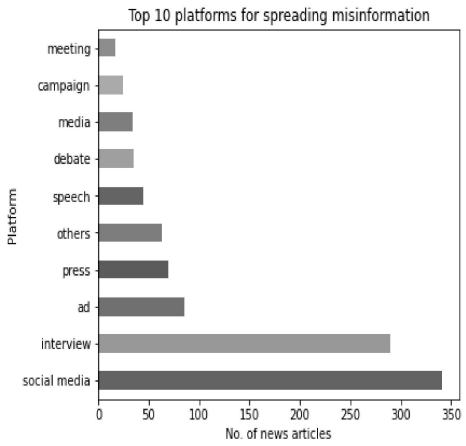

Figure 4.8: Top 10 platforms of misinformation in the dataset

As per our dataset, social media was the popular media platform for spreading misinformation followed by interviews and advertisements.

9. Visualizing the top 10 platforms for true information

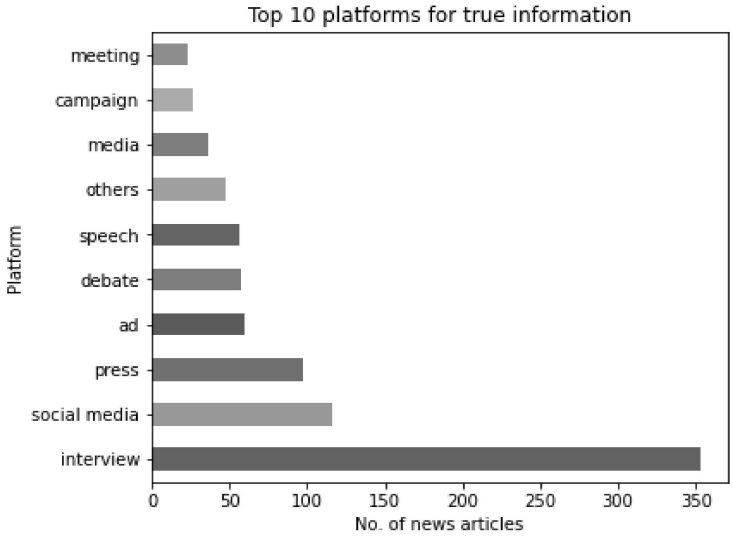

Figure 4.9: Top 10 platforms of true information in the dataset

As per our dataset, interviews was the popular media platform for true information followed by social media and press.

10. Visualization of distribution of misinformation throughout the years

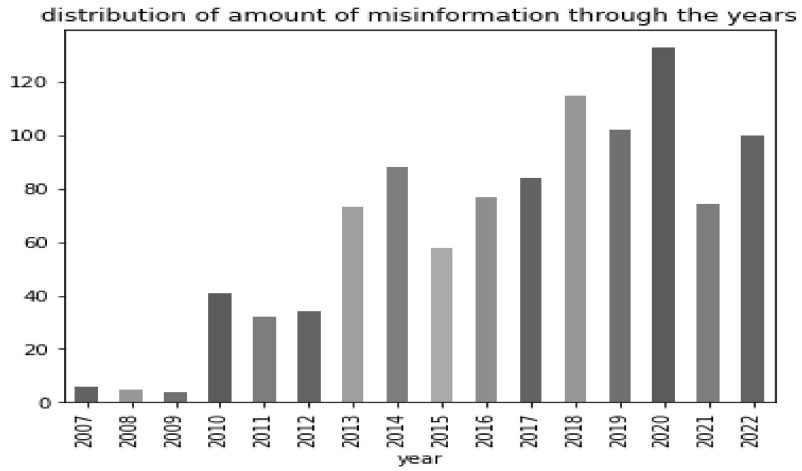

Figure 4.10: Distribution of amount of misinformation through the years

From the dataset we can infer that the year 2020 has the highest number of false statements followed by 2018 and 2022 and 2019. We can infer that the spread of misinformation goes up in years when there is an election.

11. Visualization of distribution of true information throughout the years

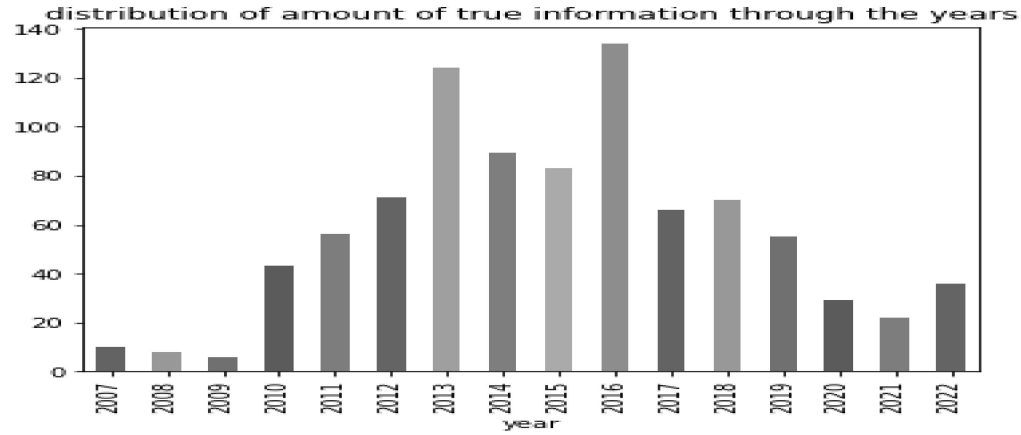

Figure 4.11: Distribution of amount of true information through the years

From the dataset we can infer that the year 2016 has the highest number of true statements followed by 2013 and 2014. It can be observed that the amount of true information has declined significantly since 2016.

12. Visualizing the top 10 fact checkers

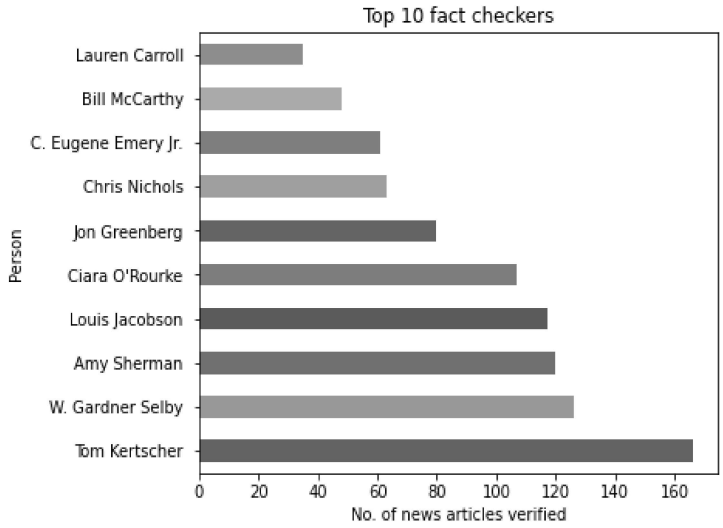

Figure 4.12: Top 10 fact checkers in the dataset

Tom Kertscher validated most of the information followed by W Gardner Shelby and Amy Sherman.

5.2 Statistical Analysis:

In this section we are checking the dependencies of categorical variables on Label a using chi-square analysis to test and come to a conclusion.

1. To check the dependency between the source and the label.

   H0(Null Hypothesis): The Label and Source are independent.

   H1(Alternate Hypothesis): The Label and Source are dependent.

```
Dependent (reject H0)
the p value is:  9.504909809381835e-19
```

Figure 5.2.1: Label and source dependency hypothesis

Result: We reject the null hypothesis as the p value is less than 0.05

2. To check the dependency between the platform and the label.

   H0(Null Hypothesis): The Label and platform where the information was posted are independent.

   H1(Alternate Hypothesis): The Label and platform where the information was posted are dependent.

```
Dependent (reject H0)
the p value is:  9.504909809381835e-19
```

Figure 5.2.2: Label and platform dependency hypothesis

Result: We reject the null hypothesis as the p value is less than 0.05

3. To check the dependency between the negative sentiment and the label.
   H0(Null Hypothesis): The Label and negative sentiment are independent.
   H1(Alternate Hypothesis): The Label and negative sentiment are dependent.

```
Independent (H0 holds true)
the p value is:  1.0
```

Figure 5.2.3: Label and negative sentiment dependency hypothesis

Result: We accept the null hypothesis as the p value is greater than 0.05

4. Top words for True and False labels.

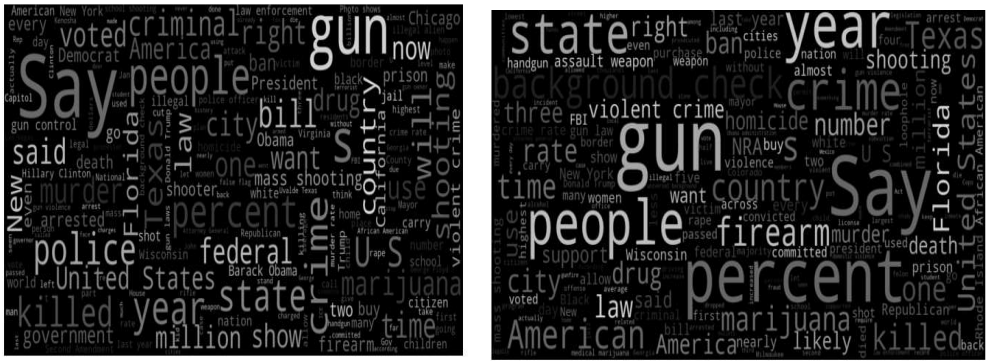

Figure 5.2.5 a) Words associated with False Label. b) Words associated with True Label

We are also trying to find the average numbers of days required to complete the fact checking of data and assign a label. It is this time that this research is trying to save These are the below results:

Average time required over the years to fact check the information: 17.4 days

Average time required over the years to check if the statement is True: 16.73 days

Average time required over the years to check if the statement is false: 17.99 days

CHAPTER 6

MODEL DESIGN

We use the below machine learning models for classifying if the given information is true or false.

6.1 Model Development:

6.1.1 Naive Bayes:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. It is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. The adjective Naïve says that features in the dataset are mutually independent. Occurrence of one feature does not affect the probability of occurrence of the other feature. Multinomial Naïve Bayes consider a feature vector where a given term represents the number of times it appears or very often [8]. For this research we are using a Multinomial Naïve Bayes Classifier with default parameters as our baseline model.

6.1.2 SVM:

SVM is one of the most popular Supervised Learning algorithms used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best 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 support vectors, and hence the 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 [9]. We use GridSearchCV to perform hyperparameter tuning to determine the optimal values for a given model. We define a dictionary with hyperparameters as 'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'poly', 'sigmoid'] and pass them

along with Support Vector Classifier as an input to the grid search model so that best hyperparameters are selected and fitted to the model.

6.1.3 Logistic regression:

Logistic regression is a supervised learning model that is used for predicting the categorical dependent variable using a given set of independent variables. It does this by predicting categorical outcomes. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables [10]. For this research we are initializing the Logistic regression model with default hyperparameters.

6.1.4 XGBoost:

Gradient boosting is an AI method used in classification and regression problems. It is implemented by the XGBoost library of Python. XGBoost is involved in reducing the loss function by using weak learners. Decision trees allude to pick the best-divided focuses considering Gini Impurity and so forth or to limit the loss function. The additive model is utilized to gather every one of the frail models, limiting the loss function. Trees are added each, ensuring existing trees are not changed in the decision tree [11]. For this research we define a dictionary with hyperparameters as {"subsample": [0.5, 0.75, 1],"learning_rate":[0.3, 0.1, 0.03],"n_estimators":[100,128,150]} and pass them along with XGBoost Classifier to the grid search so that the best hyperparameters are selected and fitted to the model.

6.1.5 Bi-Directional LSTM:

A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs (Long-Short Term Memory): one taking the input in a forward direction (past to future), and the other in a backwards direction (future to past). BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm [12]. Our research implements Bi-LSTM model with below specifications:

1st Layer — Embedding layer: Applies the embedding of the given size to the input sequence

2nd Layer — Bi-Directional LSTM Layer: Contains a LSTM with 16 neurons with a dropout rate of 0.4

3rd Layer — GlobalMaxPooling1D layer: down samples the input representation by taking the maximum value over the time dimension

4th Layer: Dense Layer: Connects all the outputs from previous layers to its neurons

Activation Function — Sigmoid Activation Function: This will give us the outputs in the values of 0 and 1

Loss Function — Binary Cross Entropy: Predicts the class output between 0 and 1

6.1.6 GRU:

GRU's are an improved version of standard recurrent neural networks. GRU has two gates: the Update Gate(z) which determines how much of the past knowledge needs to be passed along into the future and the reset Gate which determines how much of the past knowledge to forget because of this they can be trained to keep information for a long time [13]. Our research implements GRU model with below specifications:

1st Layer — Embedding layer: Applies the embedding of the given size to the input sequence

2nd Layer — GRU Layer: Contains a GRU with 100 neurons with a dropout rate of 0.2

3rd Layer: Dense Layer: Connects all the outputs from previous layers to its neurons

Activation Function - Sigmoid Activation Function: This will give us the outputs in the values of 0 and 1

Loss Function — Binary Cross Entropy: Predicts the class output between 0 and 1

6.2 Model Evaluation:

We use the below metrics to evaluate our models:

Accuracy:

Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{True\ Positives+True\ Negatives}{True\ Positives+True\ Negatives+False\ Positives+False\ Negatives}$$

Precision: Percentage of correct predictions of a class among all predictions for that class.

$$Precision = \frac{True\ Positives}{True\ positives+False\ Positives}$$

Recall: Proportion of correct predictions of a class and the total number of occurrences of that class.

$$Recall = \frac{True\ Positives}{True\ Positives+False\ Negatives}$$

F1 Score: F1 score is a weighted average of precision and recall.

$$F1\ Score = \frac{2*(Precision*Recall)}{Recall+Precision}$$

ROC Curve: A binary classification diagnostic plot.

Post testing the models on the test data the below are the results for our models

Model	Accuracy	Precision	Recall	F1
Naive bayes Classifier	0.67	0.67	0.67	0.67
SVM Support Vector Classifier with grid search	0.73	0.73	0.73	0.73
Logistic regression	0.71	0.71	0.71	0.71
XGboost with Grid Search	0.83	0.83	0.83	0.83
Bi-LSTM	0.64	0.65	0.66	0.66
GRU	0.61	0.61	0.61	0.61

Table 6.1: Model results

Post verifying the accuracy, precision and recall scores of all the models we conclude that XGboost classifier with grid search performed better than any other traditional ML models as well as deep learning models we select it to be our final model. We then save the model and pass a new article to predict if the information is false or true. This model can also be integrated with any platform for detecting and flagging false information.

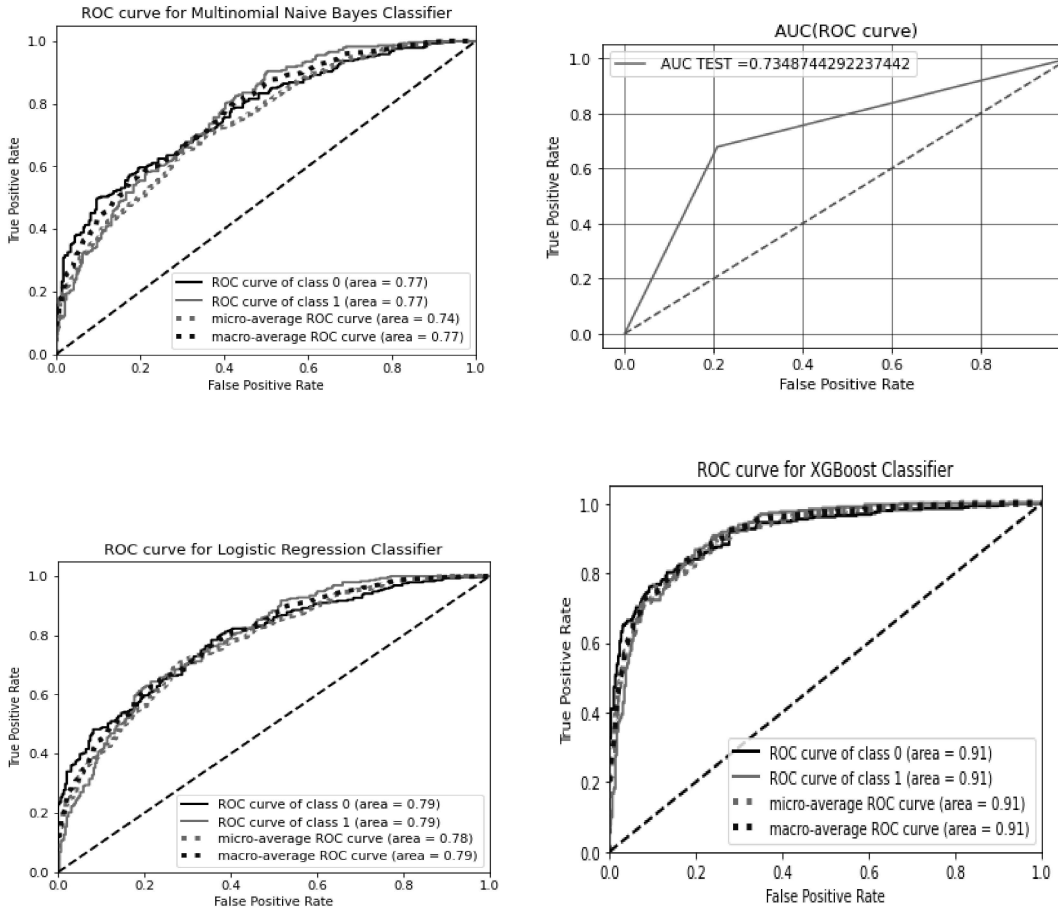

Figure 6.2.1. ROC curve for Multinomial Naive Bayes Classifier 6.2.2. ROC curve for SVM 6.2.3. ROC curve for Logistic Regression 6.2.4. ROC curve for XGBoost Classifier

From the results of our experiment, it can be inferred that the traditional machine learning models performed better than the deep learning models. Even our baseline traditional model Naïve Bayes Classifier had better accuracy and precision than the complex Bi-LSTM model. The main difference between traditional machine learning and deep learning models in this context is that, for machine learning models the applied features need to be identified manually whereas

deep learning models try to learn high-level features from data and therefore do not require hard-core feature extraction. Hence, we can assume that in our research the deep learning models were not able to learn and assign appropriate weights to the features and do require more data for better training and performance.

6.3 Discussion:

The maximum accuracy, precision, recall and the F1 score for our experiment was achieved by the XGBoost classifier which is comparable to the accuracy achieved by the XGBoost classifier proposed by [10]. Our model also excelled the performance of all the classifiers discussed in [5], where the maximum achieved accuracy was 27%. While our Naïve Bayes Classifier had an accuracy of 67% and performed worse than [9], which had an accuracy of 75%.

The deep learning models created using in [15] and [13] achieved an accuracy of more than 80%, but our deep learning models were not able to reach this performance. The reason can be that those models were trained and evaluated on publicly available datasets that were thoroughly cleaned and refined, whereas our models were trained and evaluated on customized data and would require further tuning. The other models also used Word2Vec and GloVe embeddings for feature extraction, which might have contributed to the high accuracy of the model. However, our initial models did not see much improvement in the accuracy of the models even after using the Word2Vec embeddings.

From our statistical analysis we determined that the average number of days required to fact check and assign the appropriate label by the journalists was 17.4 days. Our model takes a few seconds to read the information, perform feature extraction and predict the appropriate label. This saves days of manual labor and gives instant feedback which can reduce the speed at which misinformation is spread.

We are also only using the article as an input to the model and no other features like the stated platform or the source originator were used for predicting the label. The reason for not including them is because they might result in a biased model. We also tried to check if sentiment can help in classification.However, we were able to establish that there was no dependency between the sentiment and label and is not helpful in predicting the label of the given information.

CHAPTER 7

LIMITATIONS

7.1 Limited words:

We trained the model with the data scraped from the “Politifact” website. So the number of words or sentences that are feeded to the model are limited compared to the models like BERT that are trained on millions of words. But our project focussed on 3 main categories, i.e crime, guns and marijuana. Training the limited data using *BERT* wouldn’t add much value for this project and also its resource intensive. So the developed model is trained with a limited number of words.

7.2 Multiple Labels:

The truthfulness of the information was categorized in the ‘Politifact’ website into different classes like *mostly-true, half-true, half-false* etc. To avoid multi classification we have converted the *mostly-true* labels to *true* and *mostly-false* labels to *false*. This has been done to avoid multi-class in knowing the veracity of the information feeded to the model. It is assumed that mostly-true and mostly-false labels would always be true and false. If the legitimacy of the information or news was provided in a binary classification i.e either true or false, the model would have been performed better with no assumptions.

7.3 Crime Specific:

As mentioned above, we did our work on 3 main categories of crime. So the model is trained with the information or statements that relate to the crime. So the developed model may not predict the statements correctly that don't relate to the crime.

CHAPTER 8

FUTURE SCOPE

In this project, we have developed a model that can predict whether the given news is fake or not. With the current digital media platforms and other sources it would be easier to spread or share the news in a short time. It would be more effective and useful if we can identify them before it spreads across multiple sources or sites. In order to achieve that there is a necessity of a software plugin or extension that can be added to a web browser. The developed model should be integrated with any social media platforms or dependent websites. The model should be allowed to scan the information content **FLAG** the news item. This may warn the user about the truthfulness of the news item or post.

For this project, we took the data from one website that labels the veracity of the news. So the news content provided to the model was actually printed by a particular website or a person. So if there is a single source of labeled data from multiple sources, there would be more content to train which could be more efficient.

CONCLUSION

In this research, we first start with the importance and definitions of automatic fake news detection. We discussed what we wish you accomplish and the questions we have before starting the experiment.We start by collecting the data from a reliable source and cleaning the data so that is more readable.We also determined that the originator and platform where the information was posted do help in determining if the information is true or false.Our results show that the prediction performance of proposed features combined with existing classifiers has a useful degree of discriminative power for detecting fake news. Our best classification results can correctly detect 82% of all fake news in our data.We have also explained the shortcomings of our model and gave suggestions for our future fake news detection model.

REFERENCES

[1] B. N. A. S. a. M. R. Z Khanam, "Fake News Detection Using Machine Learning," *IOP Conference Series: Materials Science and Engineering,* 2020.

[2] S. Pappas, "Fighting fake news in the classroom," *Monitor on Psychology,* p. 53, 2022.

[3] V. V. H. a. A. Kumar, "Natural Language Processing based Online Fake News Detection Challenges – A Detailed Review," *5th International Conference on Communication and Electronics Systems (ICCES),* pp. 748-754, 2020.

[4] P. S. R. A. Shubha Mishra, "Analyzing Machine Learning Enabled Fake News Detection Techniques for Diversified Datasets," *Wireless Communications and Mobile Computing,* pp. Article ID 1575365, 18 pages, 2022.

[5] W. Y. Wang, "Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* 2017.

[6] K. a. M. D. a. W. S. a. L. D. a. L. H. Shu, "FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media," *arXiv,* 2018.

[7] W. Lifferth, "Fake News," Kaggle, 2018. [Online]. Available: https://kaggle.com/competitions/fake-news.

[8] A. J. a. A. Kasbe, "Fake News Detection," *IEEE International Student's Conference on Electrical, Electronics and Computer Science (SCEECS). Bhopal, India:,* 2018.

[9] S. Gilda, "Evaluating machine learning algorithms for fake news detection," *15th Student Conference on Research and Development (SCOReD),* pp. 110-115, 2017.

[10] A. C. F. M. A. V. a. F. B. J. C. S. Reis, "Supervised Learning for Fake News Detection," *IEEE Intelligent Systems,* vol. 34, pp. 76-81, 2019.

[11] Z. K. a. B. N. A. a. H. S. a. M. Rashid, "Fake News Detection Using Machine Learning Approaches," *IOP Conference Series: Materials Science and Engineering,* 2021.

[12] O. S. K. I. V. Jamal Abdul Nasir, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *International Journal of Information Management Data Insights,* vol. 1, 2021.

[13] K. M. S. Y. A. &. W. G. Popat, "Declare: Debunking fake news and false claims using evidence-aware deep learning," *arXiv:1809.06416,* 2018.

[14] D. B. S. Z. O. Ajao, "Fake news identification on twitter with hybrid CNN and RNN models," *Proceedings of the 9th international conference on social media and society,* pp. 226-230, 2018.

[15] Y. L. Q. X. R. L. M. a. H. C.-R. Long, "Fake news detection through multi-perspective," *In Proceedings of the Eighth International Joint Conference on Natural Language Processing,* p. 252–256, 2017.

[16] A. D. Holan, "The Principles of the Truth-O-Meter: PolitiFact's methodology for independent fact-checking," https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/, 12 February 2018. [Online].

[17] M. &. A. K. &. M. S. &. J. A. &. M. S. &. A. A. Abbas, "Multinomial Naive Bayes Classification Model for Sentiment Analysis," 2019.

[18] T. Y. X. T. Wen Zhang, "Text classification based on multi-word with support vector machine," *sciencedirect,* vol. 21, no. 8, pp. 879-886, 2001.

[19] T. P. a. V. Marcinkevičius, "Application of Logistic Regression with part-of-the-speech tagging for multi-class text classification," *016 IEEE 4th Workshop on Advances in Information Electronic and Electrical Engineering (AIEEE),* pp. 1-5, 2016.

[20] Z. Qi, "The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model," *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA),* pp. 1241-1246, 2020.

[21] N. J. a. A. -r. M. A. Graves, "Hybrid speech recognition with Deep Bidirectional LSTM,," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding,* pp. 273-278, 2013.

[22] Y. C. Y. D. J. X. a. D. P. X. Tang, "A Multi-scale Convolutional Attention Based GRU Network for Text Classification," *2019 Chinese Automation Congress,* pp. 3009-3013, 2019.