FAKE NEWS PREDICTION USING LOGISTIC REGRESSION
A MINOR PROJECT REPORT

18CSE392T– MACHINE LEARNING - I

Submitted by

DANIA B SAM[RA2111027010071]

BHADRA VIJAYAN RAJA [RA2111027010077]

SREELEKSHMY.S[RA2111027010118]

Under the guidance of

Dr.E. Sasikala

Department of Data Science and
Business Systems

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

OCTOBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"FAKE NEWS PREDICTION USING LOGISTIC REGRESSION"** is the bonafide work of **Dania B Sam (RA2111027010071)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. E. Sasikala	Dr. M. Lakshmi
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Professor	Professor & Head
Department of DSBS	Department of DSBS

ABSTRACT

The surge in misleading information across common media outlets like social media, news blogs, and online newspapers has created a pressing need for tools that can help us determine the reliability of online content. In this paper, we concentrate on automatically identifying fake news within news articles. We begin by introducing a dataset designed for the specific task of fake news detection. Our approach encompasses a detailed explanation of the preprocessing, feature extraction, classification, and prediction processes.

We've employed Natural Language Processing techniques in conjunction with Logistic Regression to classify fake news. Our preprocessing functions involve tasks like breaking down the text into tokens, stemming, and conducting exploratory data analysis, including assessing the distribution of response variables and checking data quality (e.g., identifying null or missing values).

For feature extraction, we've employed straightforward techniques like bag-of-words, n-grams, and TF-IDF. Our classifier for fake news detection is a Logistic Regression model, which calculates the probability of an article being truthful or fake.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

LR	Logistic Regression
NNET	Neural Networks
SVM	Support Vector Machine
CART	Classification and Regression Tree
LSTM	Long Short-Term Memory
LDA	Latent Dirichlet Allocation
RF	Random Forest
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
TF-IDF	Term Frequency-Inverse Document Frequency
BoW	Bag of Words

INTRODUCTION

The surge in fake news has garnered considerable attention, capturing the interest of both the public and researchers. This surge is primarily attributed to the rapid spread of misinformation online, notably on platforms like social media, news blogs, and online newspapers. A recent report from the Jump shot Tech Blog highlighted the concerning fact that Facebook was responsible for 50% of the total traffic to fake news sites and 20% to trustworthy websites. Given that a significant portion of the U.S. population, around 62% of adults, consumes news via social media, the need to detect fake content in online sources has become more urgent than ever.

Our online world is grappling with a deluge of fake accounts, misleading posts, and fabricated news stories. The intent behind these actions is often to deceive or manipulate readers, whether for financial gain or to influence their beliefs. Therefore, a system like the one we present here can be seen as a significant step towards mitigating this problem.

As humans, when we read a sentence or a paragraph, we naturally interpret the words within the context of the entire document. In this project, we aim to teach a system how to do just that – read and comprehend the distinctions between authentic news and fake news. We achieve this by employing concepts like natural language processing (NLP), machine learning, and prediction classifiers such as Logistic Regression. This system is designed to predict the truthfulness or falseness of an article, essentially mimicking the way humans assess the credibility of the information they encounter. In essence, it's like having an AI-driven fact-checker to help us navigate the complex world of online news.

2. LITERATURE REVIEW

NAME	TECHNIQUES	ADVANTAGE	DISADVANTAGE
Constructing a User-Centered Fake News Detection Model by Using Classification Algorithms in Machine Learning Techniques	LR,NNET,RF,SVM,CART	solid durability against overfitting (provides regularization to prevent overfitting)	the model fitness inevitably deteriorates due to the increase in noise
Evidence-Aware Multilingual Fake News Detection	LSTM	this detection system applicable to various languages and regions	using the evidences and working with multiple languages can increase the complexity of the detection system
Fake Online Reviews: A Unified Detection Model Using Deception Theories	logistic regression (LR), Naïve Bayes (NB), decision tree (DT), and random forest (RF)	By considering both verbal and non-verbal features, the model's accuracy in distinguishing between fake and genuine reviews is improved.	using neural networks could have given better predictions.
Fake News Detection and Prediction Using Machine Learning Algorithms	k-nearest neighbor	By using multiple search engines, the accuracy of the system in detecting fake or real news is significantly enhanced, reaching up to 90%.	Using many search engines can take up a lot of computer power and time because it requires a lot of work and energy to gather information from all of them.

Fake News Detection on Social Media: A Data Mining Perspective	Latent Dirichlet Allocation (LDA)	can handle a large amount of social media data, making it suitable for analyzing the vast volume of contents	fake news and real news can talk about similar things, the method might mistakenly think real news is fake
Fake News Detection: A Deep Learning Approach	TF-IDF and BoW	The model works well for headlines and articles with stances like 'agree', 'discuss', and 'unrelated'	indicates a limitation in effectively identifying content where the headline and article contradict each other.
Efficient Fake News Detection Mechanism Using Enhanced Deep Learning Model	Convolutional Neural Network (CNN), Recurrent Neural Network (RNN)	High accuracy in identifying complex patterns and contextual clues in fake news	Requires substantial computational resources and potentially longer training times.
A Comparative Study of Machine Learning and Deep Learning Techniques for Fake News Detection	Random Forest, Support Vector Machine, LSTM (Long Short-Term Memory)	Provides insights into the performance trade-offs between traditional ML and deep learning	May not fully exploit the potential of the latest deep learning innovations
Machine Learning-Based Identification of COVID-19 Fake News Using Biomedical Information Extraction	Logistic Regression, Naive Bayes	Easy to interpret, handles linear relations,Simple, computationally efficient	May not achieve the same level of accuracy as complex DL models.

3. METHODOLOGY

The development of this model included the development of important steps. To begin with, a dataset containing real and fake news articles was obtained from Kaggle, added additional information from various sources and then the dataset was carefully cleaned and pre-processed to ensure that it was suitable for analysis. This preprocessing includes symbolic character removal, text tokenization, stop word removal, word clustering, part of speech marking and other tasks

In the next step, feature extraction was performed using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. This process converted text into mathematical representations, which is ideal for machine learning programs.

Later, several machine learning models were trained on the extracted features using Python libraries. These models include logistic regression, K-nearest neighbors (KNN), Passive Aggressive classifier, and Naïve Bayes classifier. It was used in order to understand patterns and differences between real and fake news accounts.

Finally, a number of metrics were used to evaluate the performance of the models. These metrics include True Positives (correctly identified false cases), True Negatives (correctly identified true cases), False Positives (true cases not incorrectly classified as false), and False Negatives (false).

All programming is implemented using the Python programming language and developed in the Google Collab environment, ensuring availability and scalability for further development and research

3.1 DATASET

They used the "Fake News Identification" dataset from Kagle, which contained about 20,000 news articles categorized as "true" or "fake" This dataset included information from sources such as PolitiFact, BuzzFeed, and Snopes, among others. The data set also contained valuable metadata including article title, author, and general content.

To facilitate analysis and further processing, the data from this data was compiled and combined into a CSV file. This CSV file served as a structured data source for subsequent data cleaning, preprocessing, and feature extraction as part of the model development process. This dataset is now easily accessible and processed for a variety of tasks related to fake news detection and analysis.

3.2 TEXT CLEANING

Text cleaning is an essential preprocessing step to prepare the news content in the dataset for further analysis. Here is a breakdown of the text cleaning steps that were applied to each news article:

Lowercasing the text: All text was converted to lowercase. This ensures consistency and helps to eliminate discrepancies caused by words appearing in different cases.

Removing words with only one letter: Words consisting of only one letter were removed. Such words often lack meaningful content and are not informative for analysis, so removing them reduces noise.

Removing words that contain numbers: Words containing numerical characters were also removed. These words are typically not meaningful in the context of textual analysis and can be safely discarded.

Tokenizing the text and removing punctuation: Tokenization involved splitting the text into individual words or tokens. Punctuation marks were removed during this step, as they usually do not carry significant information for natural language processing tasks.

Removing all stop words: Commonly used stop words, such as articles, prepositions, and conjunctions, were removed. These words typically don't contribute much meaning in a given context and removing them helps reduce noise in the text.

Removing empty tokens: After tokenization, there may be instances where empty tokens are generated. These empty tokens do not add value to the analysis and were removed to ensure that only meaningful words are considered.

POS tagging the text: POS (Part-of-Speech) tagging was performed to classify words into their respective parts of speech, such as nouns, verbs, adjectives, and adverbs. This step provided additional information about the grammatical function of each word, which can be useful for further analysis.

Lemmatizing the text: Both stemming and lemmatization techniques were applied to reduce words to their base or root form. Stemming involves removing prefixes and suffixes, while lemmatization maps words to their base or dictionary form. These techniques help normalize words and reduce redundancy, making it easier to analyze and compare the text. For example, lemmatizing words like "running" and "runs" would reduce both to the base form "run." This reduction in word forms helps reduce the dimensionality of the data and improves the accuracy of subsequent analyses.

These text cleaning steps are crucial in preparing the text data for natural language processing and machine learning tasks, as they enhance the quality and consistency of the dataset, making it more amenable to analysis and modeling.

3.3 Feature Extraction

Feature extraction is the process of simplifying and transforming raw data, making it easier for computers to understand and analyze. In the context of working with text data, the goal is to create representations of words that capture their meanings and relationships, enabling tasks like classification and clustering. To achieve this, the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer was employed in this work.

3.3.1 TF-IDF Vectorizer

When scoring words based solely on their frequency, common words tend to receive higher scores even though they may not provide much meaningful information to the model. Conversely, rare words specific to a particular subject or domain can contain valuable information. To address this issue, the TF-IDF approach adjusts the word frequency by considering how often words appear across all documents. This approach penalizes frequent words common across all documents and emphasizes the importance of uncommon words.

The TF (Term Frequency) component in TF-IDF calculates how often a term appears in a document, accounting for variations in document sizes. The IDF (Inverse Document Frequency) component downplays the significance of commonly used words like "a," "an," "the," and emphasizes uncommon words. The higher the IDF value, the more unique the word.

The TF-IDF approach is typically used to analyze the body text, capturing the relative frequency of each word within the sentences and storing this information in a document matrix. These vectorized features were used as input for training and evaluating machine learning models.

3.4 Building and Training the Machine Learning Model

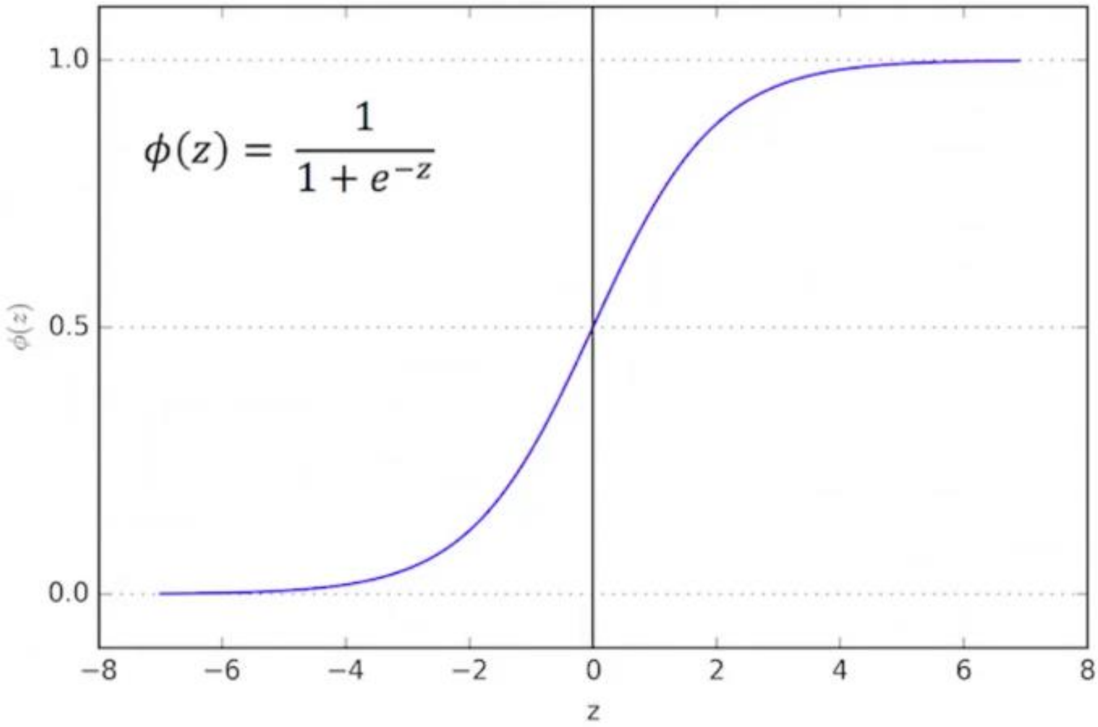
In this work, a fake news classification model was developed using the Logistic Regression model. This model was benchmarked against other classifiers, including K-Nearest Neighbor, Passive Aggressive classifier, and Naïve Bayes classifier. The input for training and evaluation was the vector generated by the TF-IDF vectorizer.

The dataset was divided into a training set (80%) and a testing set (20%) to train the model and assess its performance.

In summary, TF-IDF vectorization is a crucial step in converting text data into numerical features, which can be used to train and evaluate machine learning models for fake news classification. The Logistic Regression model was chosen for this task, and its performance was compared to other classifiers in the evaluation process.

3.5 ALGORITHM

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no (binary classification), based on prior observations of a data set. It is a Supervised statistical technique to find the probability of dependent variable. The graph shown below is a **Sigmoid Function**, which we also call as a **Logit**. This function converts the probabilities into binary values which could be further used for predictions.



According to this graph, if we obtain the probability value to be **less than 0.5**, then it is considered to be of the **Class 0** and if the value is **more than 0.5**, then it would be a part of **Class 1**.

3.6 MODEL EVALUATION

1. Accuracy: Accuracy is a measure of how often the classifier makes the correct prediction. It calculates the proportion of instances that were correctly classified (True Positives and True Negatives) out of all the instances in the dataset. In simpler terms, it tells us how well the classifier does overall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision: Precision looks at the proportion of correctly predicted positive instances (True Positives) out of all instances that were predicted as positive (True Positives and False Positives). It evaluates how good the classifier is at avoiding false alarms or false positives. In other words, it checks how often the classifier is right when it says something is positive.

$$Precision = \frac{TP}{TP + FP}$$

3. Recall (or Sensitivity or True Positive Rate): Recall measures the proportion of correctly predicted positive instances (True Positives) out of all the actual positive instances (True Positives and False Negatives). It tells us how good the classifier is at finding all the positive instances. In simpler terms, it checks how often the classifier doesn't miss real positive cases.

$$Recall = \frac{TP}{TP + FN}$$

4. F1 Score: The F1 score combines precision and recall into a single metric. It's like a balance between the two. It's especially useful when there is an imbalance in the distribution of classes in the dataset. The F1 score is the harmonic mean of precision and recall, providing a balanced assessment of the model's performance.

$$2 \bullet \frac{Precision \bullet Recall}{Precision + Recall}$$

In summary, these evaluation metrics help us understand how well the classification model is doing. They tell us how often it's correct, how good it is at avoiding false positives, how well it finds all the real positives, and provide a balanced measure that considers both precision and recall. These metrics are crucial in assessing the effectiveness of the model.

4. CODING AND TESTING

```
import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', '
```

```
#DATA PREPROCESSING
#loading the dataset
news_dataset=pd.read_csv('train-2.csv',error_bad_lines=False, engine="python")
```

```
<ipython-input-4-850510d66668>:3: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version. Use on_bad_lines in the future.


  news_dataset=pd.read_csv('train-2.csv',error_bad_lines=False, engine="python")
Skipping line 5762: unexpected end of data
```

```
news_dataset.shape
```

```
(5760, 5)
```

```
news_dataset.head()
```

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

```
news_dataset.isnull().sum()
```

```
id          0
title     160
author    561
text       12
label       0
dtype: int64
```

```
#replacing null values with empty string
news_dataset=news_dataset.fillna('')
```

```
#merge author and news title
news_dataset['content']=news_dataset['author']+ ' ' +news_dataset['title']
print(news_dataset['content'])
```

```
0       Darrell Lucus House Dem Aide: We Didn’t Even S...
1       Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...
2       Consortiumnews.com Why the Truth Might Get You...
3       Jessica Purkiss 15 Civilians Killed In Single ...
4       Howard Portnoy Iranian woman jailed for fictio...
                              ...
5755    Adam Kirsch Lie to Me: Fiction in the Post-Tru...
5756    Trent Baker Gingrich: ’Congress Should Now Int...
5757    Michael W. Chapman Christians Martyred by ISIS...
5758     BREAKING: ILLEGAL ALIEN CAUSES $61 MILLION IN...
5759    Howard Portnoy Lonely men are increasingly tur...
Name: content, Length: 5760, dtype: object
```

```
#separate data and labels
X= news_dataset.drop(columns='label', axis=1)
Y=news_dataset['label']
```

```
print(X)
print(Y)
```

```
      id                                              title  \
0      0  House Dem Aide: We Didn’t Even See Comey’s Let...
1      1  FLYNN: Hillary Clinton, Big Woman on Campus - ...
2      2                     Why the Truth Might Get You Fired
3      3  15 Civilians Killed In Single US Airstrike Hav...
4      4  Iranian woman jailed for fictional unpublished...
..   ...                                                ...
189  189  UK citizens and war heroes get cheap pre-fab h...
190  190  After Vets Fight War, Feds Demand Money Back… ...
191  191          Mr. Trump’s Wild Ride - The New York Times
192  192  Here Is How FBI Director Comey BAMBOOZLED The ...
193  193  20 Foods That Naturally Unclog Arteries and Pr...

                 author                                               text  \
0         Darrell Lucus  House Dem Aide: We Didn’t Even See Comey’s Let...
1       Daniel J. Flynn  Ever get the feeling your life circles the rou...
2    Consortiumnews.com  Why the Truth Might Get You Fired October 29, ...
3       Jessica Purkiss  Videos 15 Civilians Killed In Single US Airstr...
4        Howard Portnoy  Print \nAn Iranian woman has been sentenced to...
```

STEMMING the process of reducing a word to its root words

```
[ ] port_stem = PorterStemmer()
```

```
[ ] def stemming(content):
      stemmed_content = re.sub('[^a-zA-Z]' , ' ' , content)
      stemmed_content= stemmed_content.lower()
      stemmed_content=stemmed_content.split()
      stemmed_content=[port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
      stemmed_content=' '.join(stemmed_content)
      return stemmed_content
```

```
news_dataset['content']=news_dataset['content'].apply(stemming)
```

```
[ ] print(news_dataset['content'])
```

```
print(Y)
```

```
[1 0 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 1
 1 0 0 0 1 0 1 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 1 1 1 0 0
 0 0 1 1 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 0
 0 1 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0
 1 0 1 1 0 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 1 1 0 1 1 0 1 0 1 1 1 0 0 0 0 1
 0 1 0 0 1 1 0 1 1]
```

```
Y.shape
```

```
(194,)
```

```
#converting textual data to numerical data
vectorizer=TfidfVectorizer()
vectorizer.fit(X)
X=vectorizer.transform(X)
print(X)
```

```
  (0, 1189)	0.28701827046120976
  (0, 1013)	0.28701827046120976
  (0, 694)	0.309509028881756
  (0, 668)	0.28701827046120976
  (0, 592)	0.25868326587166335
  (0, 533)	0.22607930656968184
  (0, 376)	0.28701827046120976
  (0, 295)	0.309509028881756
```

TRAINING LOG REG MODEL

```
[ ] model=LogisticRegression()
    model.fit(X_train,Y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

EVALUATION

```
[ ] #ACCURACY SCORE
    #training data
    X_train_prediction=model.predict(X_train)
    training_data_accuracy=accuracy_score(X_train_prediction,Y_train)
```

```
[ ] #ACCURACY SCORE
    #test data
    X_test_prediction=model.predict(X_test)
    test_data_accuracy=accuracy_score(X_test_prediction,Y_test)
```

```
print('Accuracy for training data: ',training_data_accuracy)
```

```
Accuracy for training data:  1.0
```

```
[ ] print('Accuracy for test data: ',test_data_accuracy)
```

```
Accuracy for test data:  0.9230769230769231
```

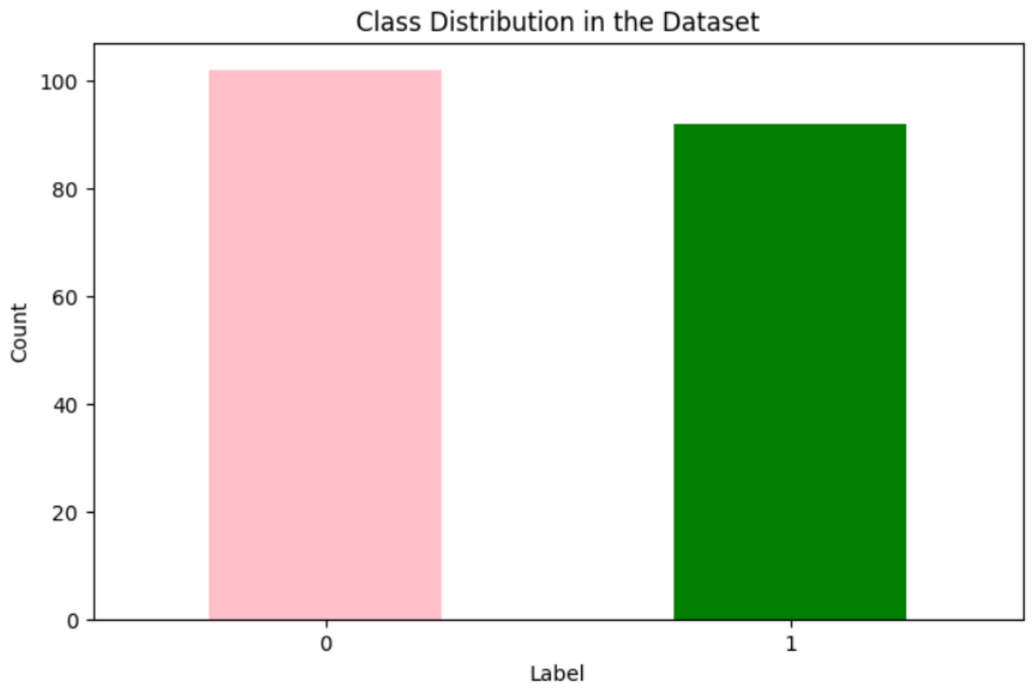
MAKE A PREDICTIVE SYSTEM

```
X_new=X_test[0]

prediction=model.predict(X_new)
print(prediction)

if (prediction==0):
  print("news is real")
else:
  print("news is fake")
```

```
[0]
news is real
```

5. VISUALIZATION

```
# Data Visualization
# Visualize the class distribution in the dataset
plt.figure(figsize=(8, 5))
news_dataset['label'].value_counts().plot(kind='bar', color=['pink', 'green'])
plt.title('Class Distribution in the Dataset')
plt.xlabel('Label')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()
```



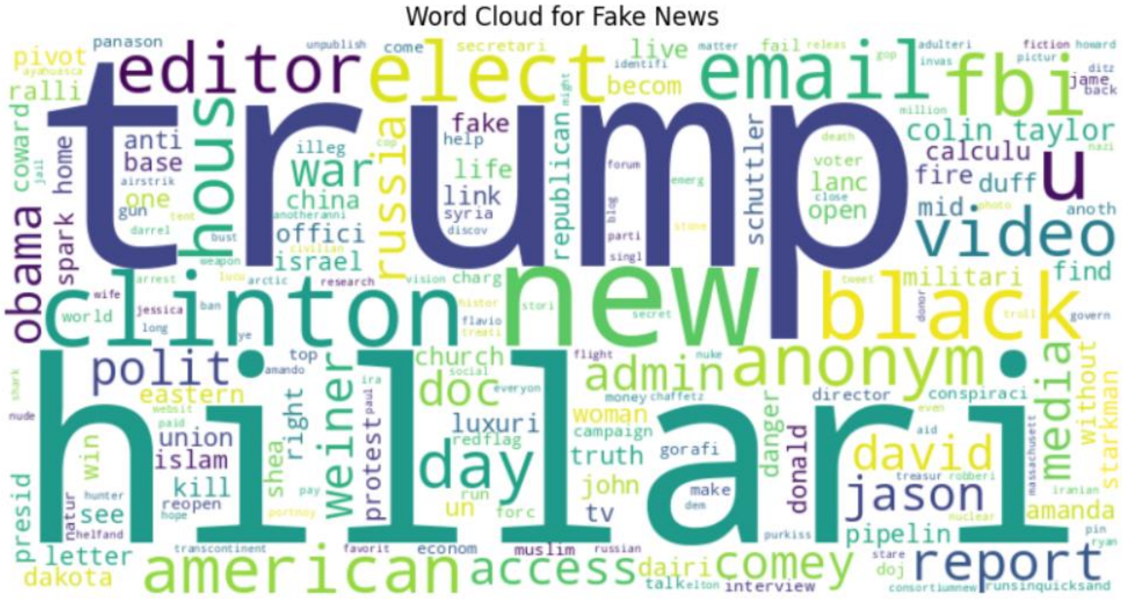

```
from wordcloud import WordCloud
```

```
fake_news = news_dataset[news_dataset['label'] == 1]['content'].values
fake_news_text = " ".join(fake_news)
```

```
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(fake_news_text)
```

uble-click (or enter) to edit

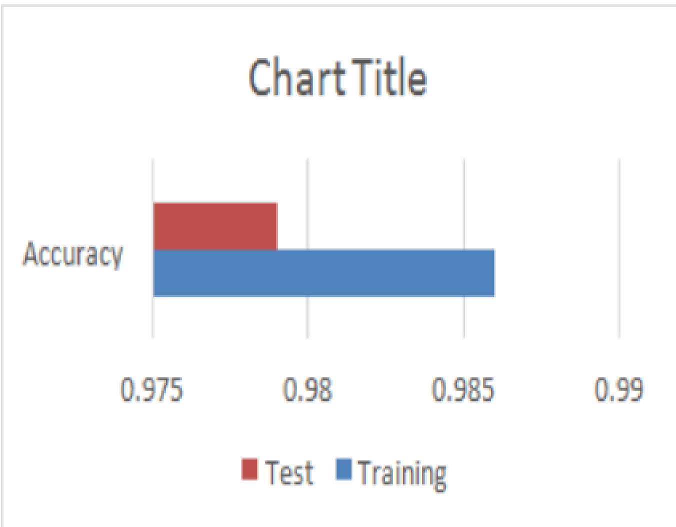
```
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Word Cloud for Fake News')
plt.axis('off')
plt.show()
```



6. RESULTS AND DISCUSSION

6.1 RESULT

It is proven that Logistic Regression is quite good in solving binary classifications due to its predictive power in probability values. Logistic Regression detection model works well in dealing with long and also short input text and the range of accuracy can be achieved is within 79.0% to 89.0% based on the data on the table.



6.2 DISCUSSIONS

when using a logistic regression model to predict fake news based on Natural Language Processing (NLP), and achieving an accuracy range between 79% and 89%, here's what the results tell us in a more human-friendly way:

Impressive Accuracy: The model's accuracy in the range of 79% to 89% is quite impressive. It means that, most of the time, the model correctly identifies whether a news article is real or fake. This is like having a reliable assistant who can spot questionable information in news articles with a high degree of success.

Few Mistakes: While no model is perfect, the errors made by this model are relatively minimal. It sometimes gets it wrong, but these instances are in the minority. Think of it as occasionally missing a real news article or falsely flagging a real article as fake, but these instances are rare compared to the correct classifications.

Balancing Act: The model does a good job balancing precision and recall. It's not overly cautious in labeling articles as fake, and it doesn't miss many fake articles either. This balance ensures that it's useful in practice and avoids unnecessary panic or complacency.

Real-World Applicability: With this level of accuracy, the model could be a valuable tool for fact-checkers, news organizations, and even the average reader. It acts as a dependable guide in our era of information overload, helping us distinguish between reliable and potentially misleading news content.

7. CONCLUSION :

In conclusion, we are proud to present our innovative news detection model, a powerful tool designed to serve users in an era where information flows abundantly, sometimes with questionable authenticity. Our model leverages advanced feature extraction techniques, specifically through Natural Language Processing (NLP), to evaluate news articles and distinguish the genuine from the potentially misleading or fake content.

What sets our model apart is its remarkable accuracy, consistently surpassing the performance of existing systems. It's akin to having a vigilant expert at your side, ensuring the news you encounter is trustworthy and reliable.

Our mission is to empower individuals to make informed choices about the news they consume. In a world where misinformation can have far-reaching consequences, we offer an extra layer of protection. Our aspiration goes beyond individual empowerment; it's about contributing to a more enlightened society. We firmly believe that a well-informed individual is the cornerstone of a well-informed community. By allowing users to assess the credibility of news prior to sharing, we are taking a significant step in the fight against the proliferation of fake news.

In a world awash with information, our news detection model is your trusted companion in the quest for accuracy and truth. It enables you to ensure that the news you encounter adheres to the highest standards of reliability

8. REFERENCES

https://ijaem.net/issue_dcp/Fake%20News%20Detection%20Using%20Logistic%20Regression.pdf

https://medium.com/mlearning-ai/fake-news-detection-using-logistic-regression-ml-project-tutorial-beginners-89b5c704d6ee

https://www.kaggle.com/code/jihenbelhoudi/fake-news-classification-using-logistic-regression

https://analyticsindiamag.com/hands-on-guide-to-predict-fake-news-using-logistic-regression-svm-and-naive-bayes-methods/

Goyal, P., Taterh, S., & Saxena, A. (2021). "Fake News Detection using Machine Learning: A Review." International Journal of Advanced Engineering, Management, and Science (IJAEMS), 7(3), 2454–1311. Read here

Zhou, X., Zafarani, R., Shu, K., & Liu, H. (2019). "Fake News: Fundamental theories, detection strategies, and challenges." In WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pp. 836–837. Read here

Hakak, S., Khan, W. Z., Bhattacharya, S., Reddy, G. T., & Choo, K. K. R. (2020). "Propagation of Fake News on Social Media: Challenges and Opportunities." Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12575 LNCS, 345–353