Enhancing Information Integrity: A Deep Learning Approach to Fake News Detection

Motamarri Lokesh,
Under the supervision of *Manish Shukla,Sudha Rani*

Department Of AI & AIDS
Parul Institute of Engineering & Technology Faculty of Engineering & Technology
Parul University, Gujarat, India

ABSTRACT

In the digital age, the rapid dissemination of information presents a critical challenge in distinguishing between authentic news and misinformation. This paper explores the development and evaluation of a deep learning model, utilizing TensorFlow, for detecting fake news with high accuracy. The model is trained on a diverse dataset, employing natural language processing (NLP) techniques to extract and analyze textual features that differentiate deceptive content from legitimate sources.
Our findings demonstrate that the proposed method significantly enhances detection accuracy when compared to traditional approaches, providing a scalable solution for real-time implementation across various media environments. The results emphasize the potential of machine learning to strengthen digital information ecosystems, addressing the growing threat of fake news in an increasingly interconnected world.

KEYWORDS

Deep Learning, Tf-idf, NLP , LSTM.

CHAPTER 1 INTRODUCTION

1.1 Introduction

These days fake news is creating different issues from sarcastic articles to a fabricated news and plan government propaganda in some outlets. Fake news and lack of trust in the media are growing problems with huge ramifications in our society. Obviously, a purposely misleading story is “fake news “ but lately blathering social medias discourse is changing its definition. Some of them now use the term to dismiss the facts counter to their preferred viewpoints.

The importance of disinformation within American political discourse was the subject of weighty attention , particularly following the American president election. The term ’fake news’ became common parlance for the issue, particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views. In this paper, it is seeked to produce a model that can accurately predict the likelihood that a given article is fake news. Facebook has been at the epi-centre of much critique following media attention. They have already implemented a feature to flag fake news on the site when a user sees it ; they have also said publicly they are working on to distinguish these articles in an automated way. Certainly, it is not an easy task. A given algorithm must be politically unbiased – since fake news exists on both ends of the spectrum – and also give equal balance to legitimate news sources on either end of the spectrum. In addition, the question of legitimacy is a difficult one. However, in order to solve this problem, it is necessary to have an understanding on what fake news.

Data or information is the most valuable asset. The most important problem to be solved is to evaluate whether the data is relevant or irrelevant. Fake data has a huge impact on lot of people and organizations. Since fake news tends to spread fast than the real news there a need to classify news as fake or not. In the project the dataset used is from Kaggle website where real news and fake news are in one dataset and trained with different machine learning classification algorithms to classify the news as fake or not. In this project different feature engineering methods for text data has been used like Tfidf model which is going to convert the text data into feature vectors which is sent into machine learning algorithms to classify the news as fake or not. With different features and classification algorithms we are going to classify the news as fake or real and the algorithm with the feature which gives us the best result with that feature extraction method and that algorithm we are going to predict the news as fake or real and also there will be ignoring attributes like the source of the news, whether it was reported online or in print, etc. and instead focus only the content matter being reported. Our aim to use different machine learning algorithms and determine the best way to classify news.

1.2 Aim of the Project

The aim of the project is classify the news is fake or real with the help of machine learning algorithm, text feature extraction methods for classifying news. In this project there is going to be use of deep learning in where there will be use of many methods processing and extraction methods to increase the accuracy rate and predict the news is fake or real.

1.3 Project Domain

The project domain for the present predicting fake news using machine learning algorithms. Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior. Artificial intelligence systems are used to perform complex tasks in a way that is similar to how humans solve problems. Machine learning is a branch of artificial intelligence that enables computers to learn from data and improve their performance without explicit programming. Machine learning can be applied to various domains, such as image recognition, natural language processing, recommended systems, and more .This project is to concentrate on the public reviews for the business,that are available in internet.Machine learning approaches are used in this research to analyse and forecast the results from the reviews. The data which is in huge amount hard to analyse the data, that will be easily analysed using machine learning algorithms.

1.4 Scope of the Project

The scope of the project is to increase the accuracy of the classification of the news as fake or real in where there will be use of the highly sophisticated classifying approach which is Logistic Regression algorithm where there will be use of the methods like text extraction,text pre-processing in this the data be trained, next is stemming process etc.,.All these methods try to remove the root words,white spaces etc., and then then extract the data so that the performance of the prediction be increased by that the user can get to know which news is fake or real.

CHAPTER 2 LITERATURE REVIEW

1. Minjung Park et al,(2023) aimed to construct a user-centered fake news detection model using classification algorithms in machine learning techniques. They used four ML techniques based on derived variables: SVM, RF, LR, and CART. These models were compared based on their performance rate in detecting fake news. To generalize these models and avoid overfitting or underfitting, the study performed cross-validation. Cross-validation involved dividing the entire data into training and test data sets and inputting them into the model establishment process. The results showed that the RF model had the highest performance rate of 94.1% in identifying fake news among fake and fact news. On the other hand, the NNET model had the lowest prediction rate of about 92.1%.In summary, this study aimed to construct a user-centered fake news detection model using classification algorithms in machine learning techniques. The results showed that the RF model had the highest performance rate in identifying fake news among fake and fact news. The results of this study are expected to contribute to improving the fake news detection system in preparation for the more sophisticated generation and spread of fake news.
2. Kai Shu et al,(2019) discussed that, a system to understand and detect fake news. FakeNewsTracker benefits researcher in identifying fake news by automatically collecting data for news and social context with a number of effective visualization techniques. The dataset has been built through Pontificate and twitter feed and considers article body, retweets and engagements as the features for binary classification of news article. LSTM with two layers consisting of 100 cells has been employed as their base technique to train the model and testing has been done with other Machine Learning algorithms like Support Vector Machine, Logistic Regression and Naive Bayes Classifier. While Support Vector Machine and Logistic Regression obtained relatively close accuracy's at 68.4 percent and 68.3 percent respectively, Naive Bayes returned 62 accuracy. Also, retweets were not considered for both training and testing. The experiment was performed on a crowd sourced database and not a standard dataset and accuracy's obtained are at best.
3. R. A. Monteiro et al,(2018) used two different techniques have been used for fake news detection. Both these techniques are applied at the same time for better accuracy. The main categories used are linguistic cue approaches with the help of ML and network study. A number of approaches have been applied for finding out three types of fake news. The study used multiple posts and contents shared by the users and news text.
4. Mykhailo Granik, et al,(2018) offered a simple technique for utilising a naive Bayes classifier to identify false information.. They were gathered from three significant left-, right-, and centre-of-

the-road Facebook pages as well as CNN, ABC News, Politico, and three other top mainstream political news sources. A classification accuracy of roughly 74% was attained. The categorization accuracy of false information is marginally worse. In this paper, they offer a simple technique for utilizing a naive Bayes classifier to identify false information.

5. Kai Shu, Suhag Wang et al,(2018) performed a fake news detection is only from content and not satisfactory and that also profound social compact to improve fake news detection. In the comparative analysis over implicit and explicit profile features between the user who reveals their potential to differentiate fake news.

6. Marco Luigi Della Vedova et al,(2018) proposed a ground-breaking machine learning (ML) method for spotting fake news that surpasses existing approaches and achieves an accuracy of up to 78.8. Second, they tested and validated their strategy using an actual application and a Face book Messenger Chabot, which led to an accuracy of 81.7 in identifying bogus news. They started out by going through the datasets they used to run their test to assess if a news article was true or not. The content-based strategy they used and how it was combined with a previously examined social-based strategy were then described. The following dataset has 15,500 postings spread across 32 pages (14 collusion pages and 18 logical pages), and it contains preferences from more than 2,300,000 people. Not all of the 6,577 posts and 8,923 (57.6) are fake.

7. Estee et al,(2018) trained the classifier by applying used features for bot detection in order to identify fake accounts created by the human on Twitter. The training is based on supervised learning. They have tested for three different classifiers, i.e., Support Vector Machine (SVM) with linear kernel, Random Forest (RF) and Adaboost. For SVM classification, the SVM linear library in R software is used. Here the boundary based on feature vectors is created for classification. For the RF model, the RF library in R software is used. RF model creates variations of trees, and mode of class outcome is used to predict identity deception. For boosting model, the Adaboost function in R is used. Adaboost is used along with decision trees where each feature is assigned a different weight to predict the outcome. These weights are iteratively adjusted, and output is evaluated for the effectiveness of identity deception prediction at each iteration. This process is repeated until the best result is obtained. Among these three classifiers, RF reached the best result.

8. Sardar Hamidian et al,(2015) performed the analysis with three data setsObama and Palin, and a mixed data set (MIX) that consists of all the data from selected five rumors. They considered number of parameters to achieve a better prediction accuracy like lexical features, unigrams, bigrams, parts of speech, sentiments, emoticons, replies, re-tweets, user id, hash tags and the time it was tweeted. Different levels of preprocessing such as lemmatization, removing punctuation, lowercasing and removal of stop words were applied to the contents of tweets. The authors concluded that the accuracy is not improved much from applying preprocessing, but instead may result in loss of some valuable information. Classification was done with the help of J48 Decision Tree and WEKA platform for training and testing.

9. Raveena Dayani et al,(2015) analysed Twitter data by considering parameters like the date when the tweet was posted, the user id, content of the tweet, tweet label, tweet id and some other related features. The Twitter data was collected using the Twitter Search API and stored in MySQL database. The authors used their own preprocessing algorithm before applying the classification algorithms. The algorithms used for classification include K-Nearest Neighbor (k-NN) and Na¨ıve Bayes. In case of k-NN, the Euclidian distance was calculated over the user-based features. However, the prediction accuracy for endorses was 73.8 and that for denies was as low as 40.9 . The authors justify the low accuracy of k-NN by the fact that the User based features had actually no correlation with the rumor detection. To apply Na¨ıve Bayes' algorithm the events that were considered were the word frequencies present in the tweets. Two important categories of factors that were considered were the User based factors (like the time when user created the account, the number of followers and followees, the number of tweets posted by the user and total number of favorites obtained for each user) and the Content 679 based factors. The Na¨ıve Bayes method had more prediction accuracy than the k-NN.Aditi Gupta et al,(2014) discussed the Twitter Streaming API. The two events that were considered for analysis were the Boston Marathon Blasts (2013) and the Hurricane Sandy (2012). To analyze the temporal distribution of tweets, the number of tweets posted in

each hour after the event occurred were considered. One of the important conclusions made was that fake content propagates faster than the real content and occurs much during the very beginning of the event. They not only analyzed the posted tweets but also the tweets from suspended users as well. User based features like the number of followers and following were also considered. Na¨ıve Bayes and Decision Tree were used for classification of which Decision Tree provided higher prediction accuracy of 96.65 as compared to Na¨ıve Bayes which provided a prediction accuracy of 91.52.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

There exists a large body of research on the topic of machine learning methods for deception detection, most of it has been focused on classifying online reviews and publicly available social media posts. Particularly since late 2016 during the American Presidential election, the question of determining ’fake news’ has also been the subject of particular attention within the literature. Conroy, Rubin, and Chen outlines several approaches that seem promising toward the aim of perfectly classifying the misleading articles.

They note that simple content-related n-grams and shallow parts-of-speech tagging have proven insufficient for the classification task, often failing to account for important context information. Rather, these methods have been shown useful only in tandem with more complex methods of analysis. Deep Syntax analysis using Probabilistic Context-Free Grammar has been shown to be particularly valuable in combination with n-gram methods. Feng, Banerjee, and Choi can achieve 85%-91% accuracy in deception-related classification tasks using online review corpora.

3.2 Proposed System

Fake news using machine learning algorithms in which our main goal is to identify fake news on various social media by utilizing user data and tweets,business news and to offer a comprehensive solution for identifying which algorithm gives the best accuracy in finding the fake news.

In this proposed model a model is build based on the count vectorizer or a tfidf matrix (i.e) word tallies relatives to how often they are used in other artices in your dataset) can help
. Since this problem is a kind of text classification, Implementing a Naive Bayes classifier will be best as this is standard for text-based processing. The actual goal is in developing a model which was the text transformation (count vectorizer vs tfidf vectorizer) and choosing which type of text to use (headlines vs full text). Now the 9 next step is to extract the most optimal features for countvectorizer or tfidf-vectorizer, this is done by using a n-number of the most used words, and/or phrases, lower casing or not, mainly removing the stop words which are common words such as “the”, “when”, and “there” and only using those words that appear at least a given number of times in a given text dataset.

3.3 Feasibility Study

A feasibility analysis evaluates the project's potentialfor success; therefore, perceived objectivity is an essen-tial factor in the credibility of the study for potential in-vestors and lending institutions. In the case of this re-search project, the feasibility analysis will try to outlinethe How's and Why's of this implementation and its re-quirements. Therefore the feasibility study will examine separately this study area which would result as follow.

3.3.1 Economic Feasibility

Economically, this project is completely feasible because it requires no extra financial investment and with respect to time, it's completely possible to complete this project in 3 months.
In this project,it verify which proposal is more economical. We compare the financial benefits of the new system with the investment. The new system is economically feasible only when the financial benefits are more than the investments and expenditure. Economic Feasibility determines whether the project goal can be within the resource limits allocated to it or not. It must determine whether it is worthwhile to process with the entire project or whether the benefits obtained from the new system are not worth the costs. Financial benefits must be equal or exceed the costs. In this issue, we should consider:

- The cost to conduct a full system investigation.
- The cost of h/w and s/w for the class of application being considered.
- The development tool.
- The cost of maintenance etc...

Our project is economically feasible because the cost of development is very minimal when compared to financial benefits of the application.

3.3.2 Technical Feasibility

The project fake news prediction verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not.The Google colab is feasible for python script and gives the accurate results. Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained. The system can be feasible because of the following grounds:

- All necessary technology exists to develop the system.
- This system is too flexible and it can be expanded further.
- This system can give guarantees of accuracy, ease of use, reliability and the datasecurity.
- This system can give instant response to inquire.

3.3.3 Social Feasibility

In this project,it can clearly understand the new from the particular business news,social media news etc.,People will share their reviews in the socail media in the form of context based on the real time news on their own perspective. But reading every reviews is a tough thing. And also it may varies from a person to per-son. So by this project, it can overcome this problem.In this project it can easily understand all the content using text processing using machine learning algorithms and avoids some of the words so that it is beneficial to the users they can understand which news is real or fake.

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.4 System Specification

3.4.1 Hardware Specification

System : Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz Hard Disk : 1 TB. Input Devices : Keyboard, Mouse Ram : 4 GB.

3.4.2 Software Specification

Operating system : Windows XP/7/10.

Coding Language : Python Tool : Google colab

3.4.3 Standards and Policies

Google Colab

Google Colab is a cloud-based platform that provides a Jupiter notebook environment for running Python code, especially geared towards machine learning (ML) and data analysis tasks. Available universally via a web browser, Google Colab supports multiple operating systems including Windows, Linux, and MacOS. It integrates seamlessly with google drive for storage and collaboration, enabling users to share and access notebooks easily. Google Colab offers a variety of built-in libraries and frameworks specifically for ML projects, such as TensorFlow, PyTorch, and scikit-learn, enhancing the ease of coding and deployment. Additionally, the platform provides free access to GPUs and TPUs, thereby accelerating computationintensive processes. This, along with its interactive UI, makes it an invaluable tool for both novice and advanced practitioners in the field of machine learning.
Standard Used: ISO/IEC 27001

CHAPTER 4 METHODOLOGY

4.1 General Architecture

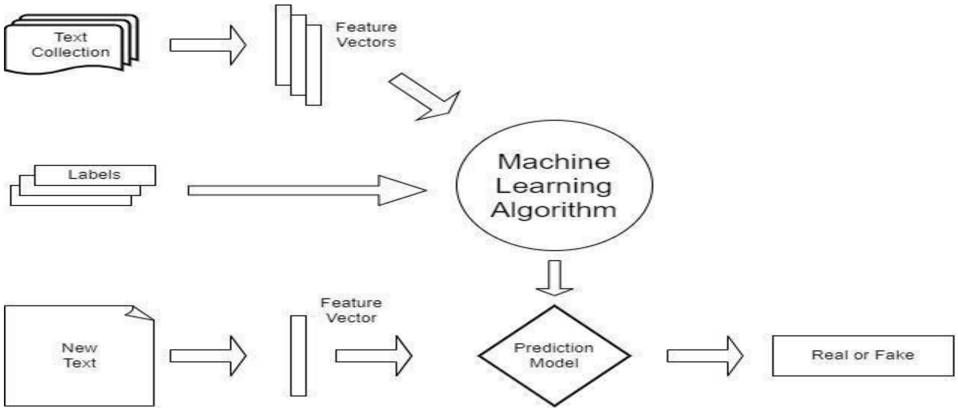


Figure 4.1: Architecture Diagram

In the figure 4.1 The image is a simple flowchart that illustrates the process of classifying news articles as fake or real. It begins with ”Text Classification” at the top, followed by ”Text Pre- processing,” then ”Feature Extraction,” and ”Classifier.” while the classifier uses the features to make predictions about the classification of the news articles. The flowchart also includes ”News to be classified” feeding into the process, with the final outcome being to ”Classify news as Fake or Real.” The flowchart presents a structured approach to determining the veracity of news content using text analysis.

4.2 Design Phase

4.2.1 Data Flow Diagram

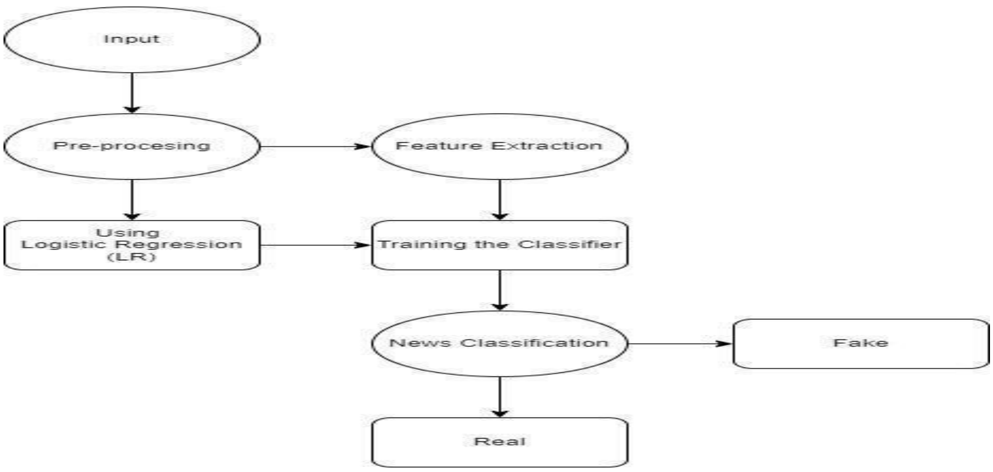


Figure 4.2: Data Flow Diagram

In the figure 4.2,the data flow diagram outlines the process of news classification, specifically distinguishing between ”fake” and ”real” news, using logistic regression as the classifier. The process begins with input, which is then pre-processed to clean and format the data appropriately. Following pre-processing, feature extraction is performed to derive meaningful features from the data that will be used to train the classifier. These features are then utilized in the training phase of the logistic regression classifier. Once the classifier has been trained, it can be used to classify news as either ”fake” or ”real”.

4.2.2 Use Case Diagram

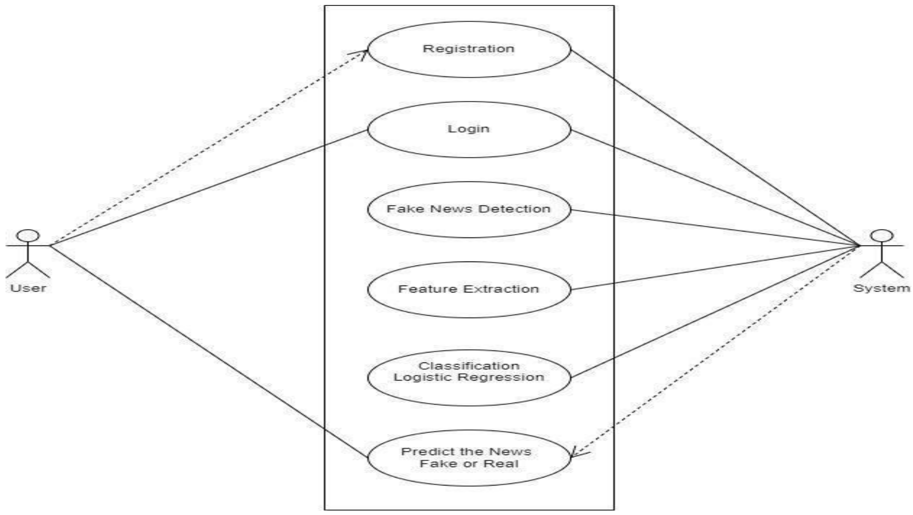


Figure 4.3: Use Case Diagram

In the figure 4.3,the use case diagram illustrates how the user interacts with the system to register, log in, submit news data for classification, and receive predictions on whether the news data is fake or real. The image describes a machine learning pipeline for news classification, specifically categorizing news as either ”fake” or ”real”. The data flow consists of several steps, starting with input, followed by pre-processing, feature extraction, and training the classifier. During pre-processing, data is cleaned and formatted appropriately. Feature extraction is then performed to derive meaningful features from the data that will be used to train the classifier. In this case, logistic regression is used as the classifier, which is trained using the extracted features. Once the classifier has been trained, it can be used to classify news as either ”fake” or ”real”.

4.2.3 Class Diagram

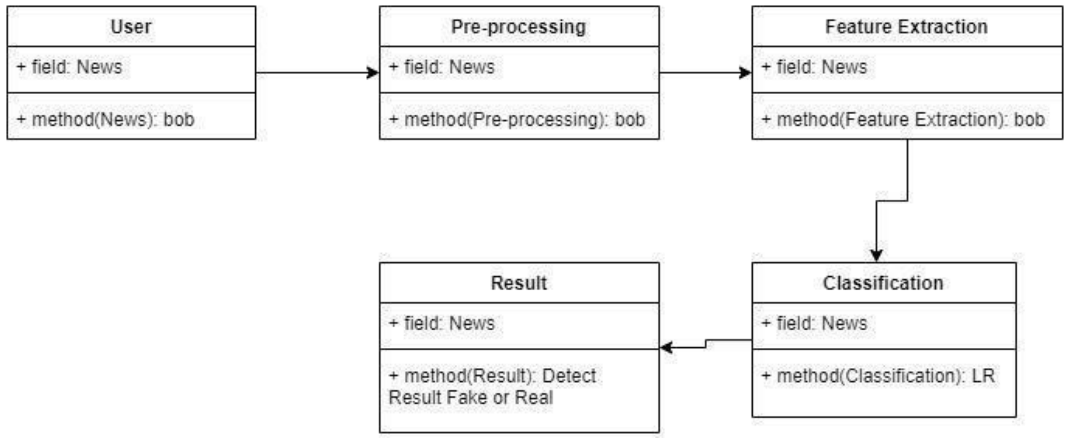


Figure 4.4: Class Diagram

In the figure 4.4, The image is a class diagram that represents a system designed to classify news as either fake or real. It consists of four main components: User, Preprocessing, Feature Extraction, and Classification, which are connected in a linear sequence. The User component has a field for News and a method that appears to be associated with a user named Bob. The Pre-processing component also includes a News field and a method for pre-processing, linked to Bob. Feature Extraction follows, with a News field and a method for extracting features, again associated with Bob. The Classification component has a News field and a method for classification using Logistic Regression (LR). Finally, there is a Result component that connects back to the Classification component, with a News field and a method to detect whether the result is fake or real. The diagram uses standard UML notation with classes represented as rectangles containing fields and methods, and arrows indicating the flow or relationship between these components.

4.2.4 Sequence Diagram

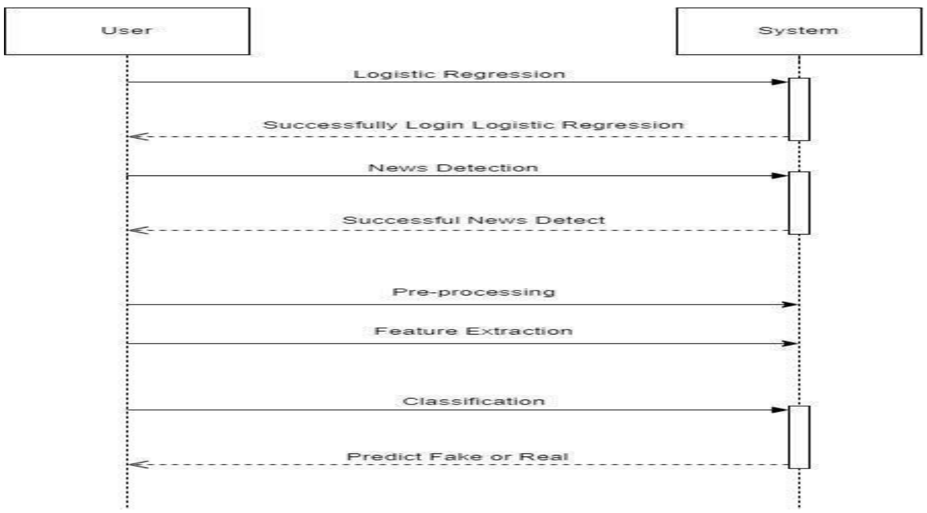


Figure 4.5: Sequence Diagram

In the figure 4.5,the sequence diagram demonstrates the flow of actions from user registration and login through submitting news data .The process consists of several steps, including input, pre-processing, feature extraction, using logistic regression, training the classifier, and news classification. During pre-processing, data is cleaned and formatted appropriately. Feature extraction is then performed to derive meaningful features from the data that will be used to train the logistic regression classifier. Once the classifier has been trained, it can be used to classify news as either ”fake” or ”real”. It shows how the user and the system interact to achieve the goal of classifying news data.

4.2.5 Activity Diagram

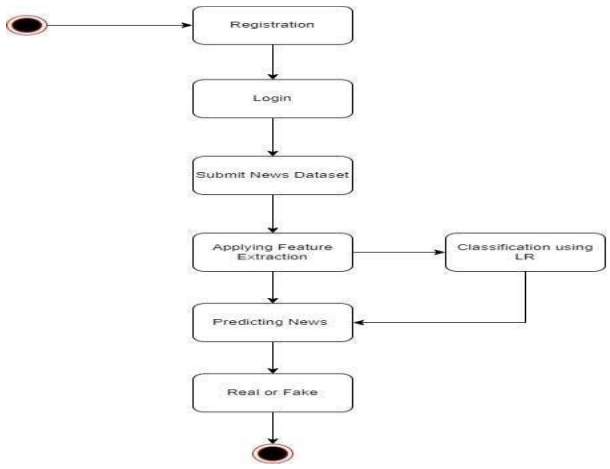


Figure 4.6: Activity Diagram

In the figure 4.6 ,the activity diagram depicts a flowchart outlining a process for handling and classifying news data. The process begins with a ”Registration” step, followed by ”Login.” Once logged in, the user submits a ”News Dataset,” which then goes through ”Applying Feature Extraction.” The extracted features are used in ”Classification using LR” (likely Logistic Regression), which feeds into ”Predicting News.” The final outcome of the process is determining whether the news is ”Real or Fake.” The flowchart uses standard symbols, with rectangles for process steps and arrows indicating the flow of the process, starting from the top and concluding at the bottom where the decision about the news’ authenticity is made.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

Step 1: Start

Step 2: Choose the Dataset

Step 3: Preparing the Dataset for Training

Step 4: Create the Training Data

Step 5: Splitting the Dataset

Step 6: Assigning the Labels and Features

Step 7: Converting the Raw Data into Numerical Data

Step 8: Split the X and Y for use in LR

Step 9: Define, compile, and train the LR model **Step 10**: Predicting the Accuracy Score of the Model **Step 11**: Stop

4.3.2 Pseudo Code

Import the necessary libraries.

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.

Downloading the Natural Language Toolkit. import nltk nltk.download('stopwords')

Printing the stopwords in English. print(stopwords.words('english'))

Data Pre-processing. news dataset = pd.read csv('/content/train.csv') Shape of the Dataset news dataset.shape

Printing the data. news dataset.head()

Counting the no of missing values in the dataset. news dataset.isnull().sum() Replacing the null values with empty string. news dataset = news dataset.fillna('') Printing the content. print(news dataset['content'])
Separating the data and label.

X= news dataset.drop(columns='label', axis=1) Y= news dataset['label']

Print XY Print(X) Print(Y)

Applying Stemming Process. def stemming(content):
stemmedcontent = re.sub('[a-zA-Z]',' ',content)ˆ stemmedcontent = stemmedcontent.lower()

stemmed content = stemmed content.split() stemmedcontent = [port stem.stem(word) for word in not stemmedcontent if word in stopwords.words('english')] stemmed content = ' '.join(stemmed content) return stemmed content

Converting the textual data to numerical data vectorizer = TfidfVectorizer() vectorizer.fit(X) X= vectorizer.transform(X)

Traing and Testing the machine learning model.

X train, X test, Y train, Y test = train test split(X, Y, test size = 0.2, stratify=Y, random state=2)

model = LogisticRegression() model.fit(X train, Y train) Accuracy of the traing data and testing data.
X train prediction = model.predict(X train) training data accuracy = accuracy score(X train prediction, Y train) print('Accuracy score of the training data : ', training data accuracy)

X test prediction = model.predict(X test) test data accuracy =

accuracy score(X test prediction, Y test) print('Accuracy score of the test data : ', test data accuracy)

Predicting the news real or fake. X new = X test[3]

prediction = model.predict(X new) print(prediction) if (prediction[0]==0):

print('The news is Real') else: print('The news is Fake')

4.4 Module Description

4.4.1 Module 1: Text Collection

The dataset was taken from Kaggle the content and metadata has been extracted from 244 web sites that have been considered to be associated with fake news by the BS Detector. Chrome Extension by Daniel Sieradski. It consists of almost 13000 posts over a period of 30 days. Research on this dataset using language processing tools has already been carried out by Kaggle users. The dataset was generated by Andrew Thompson to create document term matrices using the articles and analyse connections between articles using common political affiliations, medium or subject matter. It contains articles from top 15 American publications and the articles were mostly published between the years of 2016 and 2017. It consists of around 150000 articles that were collected by scraping news website homepages and RSS feeds. However, we will randomly select only 13000 articles from this dataset and merge it with the fake news dataset for more accurate predictions and for avoiding a skewed dataset.

4.4.2 Module 2: Text Pre-Processing

After a text is obtained, we start with text preprocessing.Text preprocessing includes:

- Converting all letters to lower case.
- Removing numbers.
- Removing punctuations, accent marks.

- Removing white spaces.
- Removing stop words.

4.4.3 Training The Models

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled ”0” and ”1”. In the logistic model, the log-odds (the logarithm of the odds) for the value labeled ”1” is a linear combination of one or more independent variables (”predictors”); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled ”1” can vary between 0 (certainly the value ”0”) and 1 (certainly the value ”1”), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio. The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier.

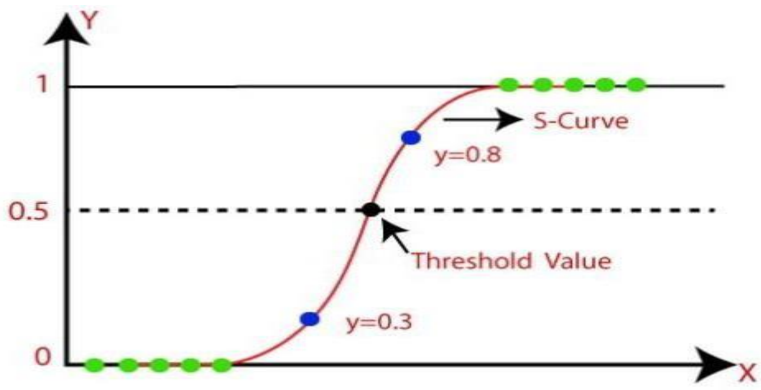


Figure 4.7: Logistic Regression

Training these models involves multiple steps:

- **Feature Extraction:** Text needs to be converted into numbers before it is used with a machine learning algorithm. For classification of documents, documents are taken as input and a class label is generated as output by the predictive algorithm. The documents need to be converted into fixed-length vectors of numbers for the algorithm to take them as input.The input for the machine learning algorithm are the words encoded as integers or floating point values.

- **Count Vectorizer:** Count Vectorizer generates an encoded vector that contains the length of the entire vocabulary coupled with the frequency of each word bywhich it appears

in the document.

- **Term Frequency-Inverse Document Frequency (TF-IDF):** Term Frequency (TF) = (Number of times term t appears in a document)/(Number of terms in the document)

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse Document Frequency (IDF) = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in. The IDF of a rare word is high, whereas the IDF of a frequent word is likely to be low. Thus having the effect of highlighting words that are distinct.

$$idf(w) = log(\frac{N}{df_t})$$

We calculate TF-IDF value of a term as = TF * IDF.

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

– tf = number of occurances of i in j

– df = number of documents contaning i

– N = total number of documents

- **Word Embedding:** It is a representation of text where words that have the same meaning have a similar representation. In other words it represents words in a coordinate

system where related words, based on a corpus of relationships, are placed closer together. Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector. Each word is represented by a real-valued vector.

- **Classifier:**The feature vectors are send to the classifier to classify the news as fake or not.

- **Accuracy:Formula**

Accuracy = TP + TN/TP + TN + FP + FN

-TP = True positive

-TN = True negative

-FP = False positive

-FN = False Negative.

4.5 Steps to Implement the project

Step 1: In first step, we have extracted features from the already pre-processed dataset. These features are: Tf-Idf Features .

Step 2: In this step it, built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers then it make use of Logistic Regression classifiers from sklearn. Each of the extracted features was used in all of the classifiers.

Step 3: Once fitting the model, then compared the f1 score and checked the confusion matrix.

Step 4:After fitting all the classifiers, the best performing model was selected models for fake news classification.

Step 5:In the next step it, performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier.

Step 6: Finally selected model was used for fake news detection with the probability of truth.

Step 7:Thus, finally selected and best performing classifier was Logistic Regression which was then saved on disk. It will be used to classify the fake news. It takes a news article as input from user then model is used for final classification output that is shown to user along with probability of truth.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

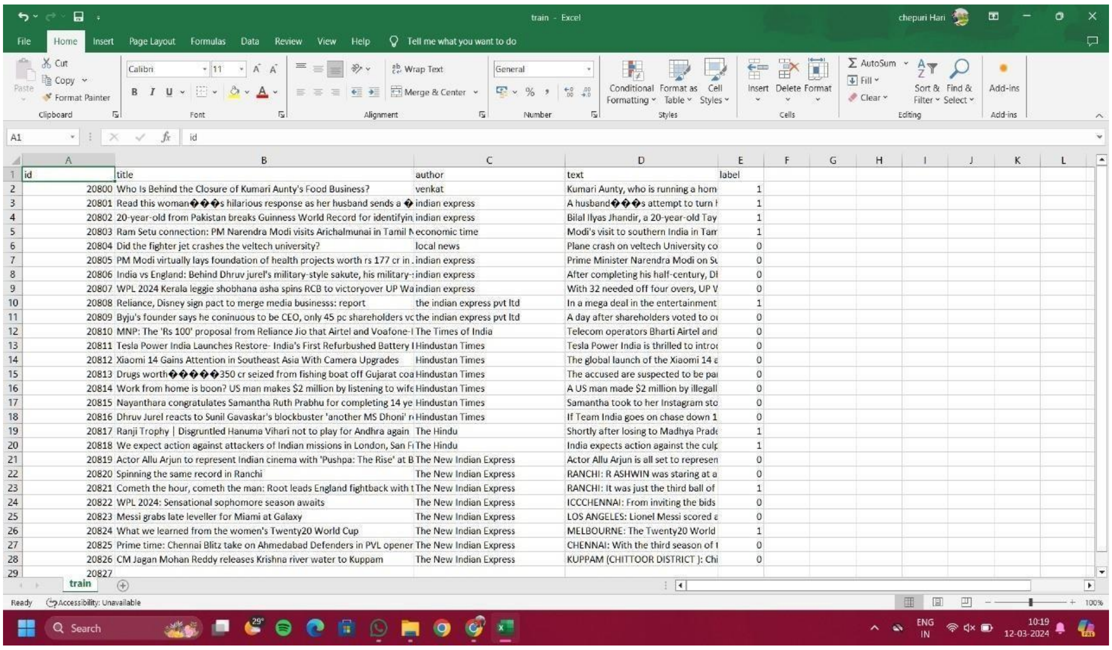


Figure 5.1: Input data

The figure 5.1, is a dataset that contains all of the information that must be analysed. The various data types included in the dataset are also regarded as input.

5.1.2 Output Design

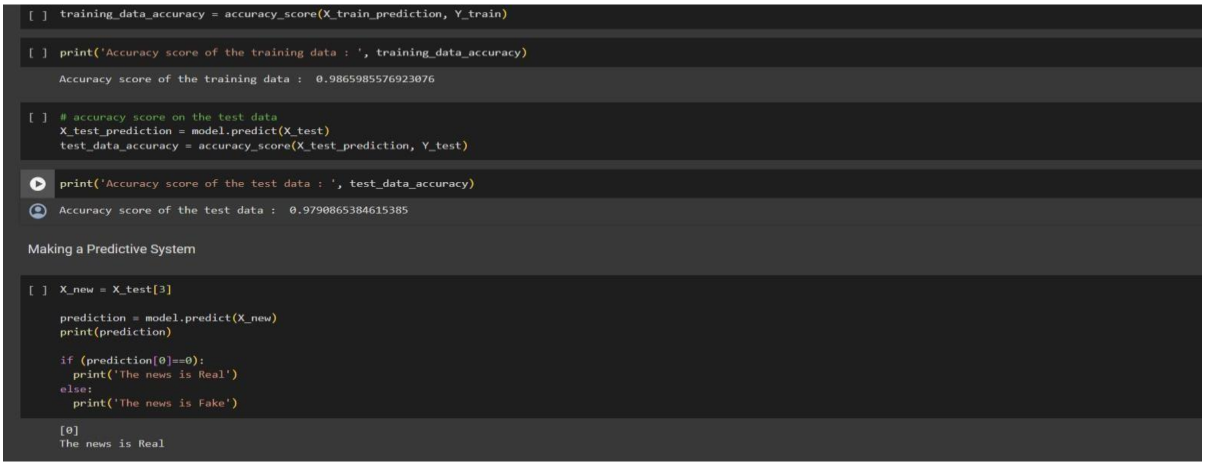


Figure 5.2: output of accuracy

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

In the figure 5.2,it shows output of fake news prediction and also it shows the accuracy and it gives that the news is real or fake.

5.2 Testing

Software testing is the process of checking the quality, functionality, and performance of a software product before launching. To do software testing, testers either interact with the software manually or execute test scripts to find bugs and errors, ensuring that the software works as expected.

5.3 Types of Testing

- Unit Testing
- Integration Testing
- System Testing

5.3.1 Unit Testing

Input

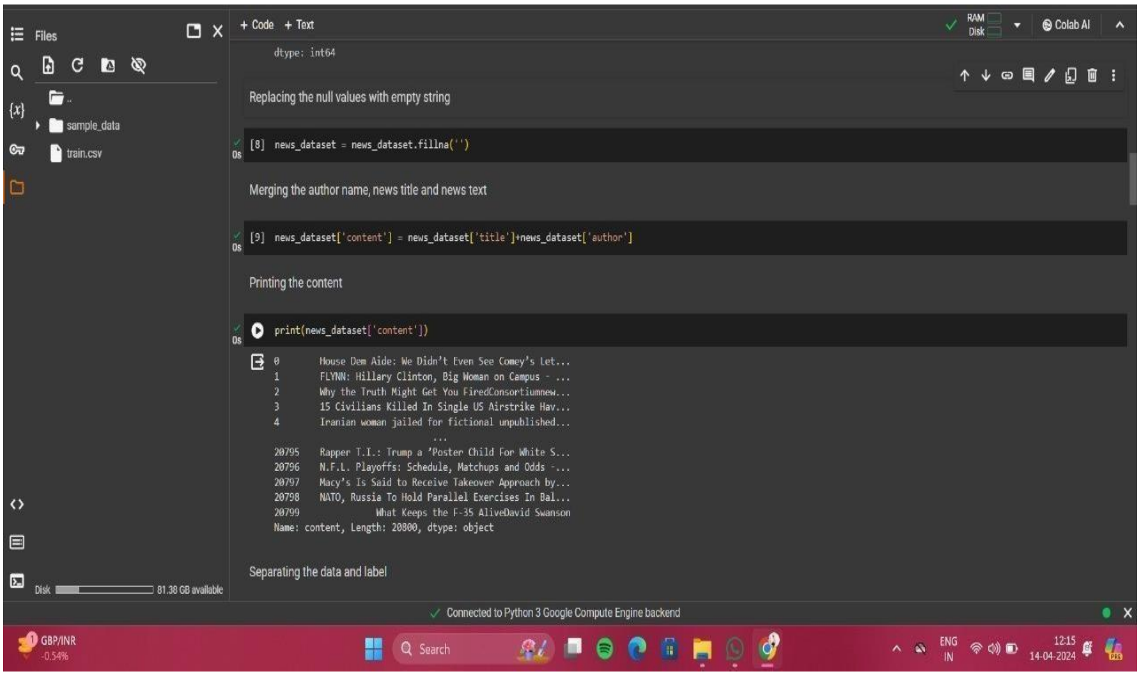


Figure 5.3: Unit Testing of Fake news prediction

In the figure 5.3 ,Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated.It is the testing of individual software units of the application.It is done after the completion of an individual unit before integration.Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.3.2 Integration testing

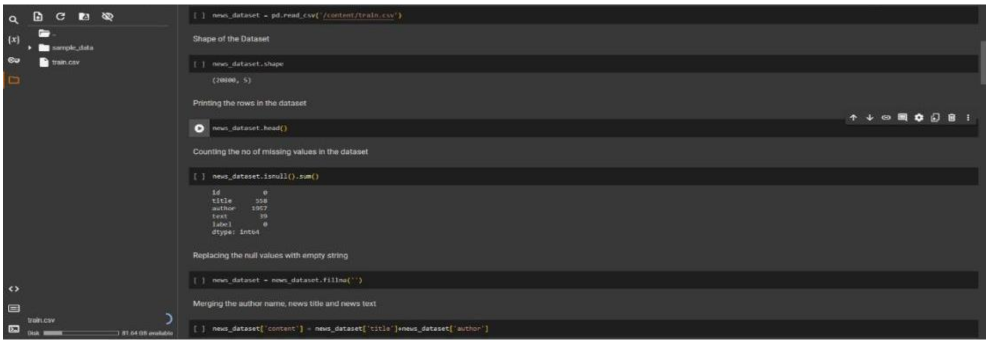


Figure 5.4: Integration Testing of Fake News Predictional

In figure 5.4, Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration testing is also called as self testing. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.3.3 System Testing

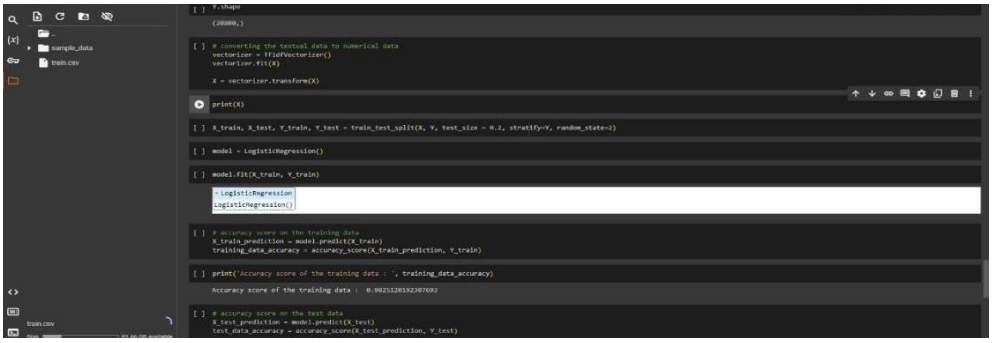


Figure 5.5: System Testing of Fake News Prediction

In figure 5.5, The purpose of System testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product.

CHAPTER 6 RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system is based on a Logistic Regression Algorithm. In that system, the accuracy of this is done using Logistic regression, which gives the output approximately 97% to 98% based on the efficiency of the data set. Logistic Regression implements many of the training models and also gives the most accurate output when compared to the other algorithms. Firstly the algorithm extracts all the text data from the dataset from the input data set and train data set. The logistic regression always includes some of the steps as follows: Selecting the training dataset, data pre-processing, stemming which is the process of reducing the root word, using of training algorithm which is logistic algorithm, and lastly evaluating the test results by the efficiency of the output will be accurate with an accuracy rate of 98%.

6.2 Comparison of Existing and Proposed System

Existing system:

In the existing system, there is be use of many Machine Learning algorithms that predict whether real or fake. When using these models, it gives the training dataset accuracy that keeps improving with the splits. The advantage of the existing system it predicts the output efficiently but the accuracy compared to all the models the logistic regression has less accuracy.

Proposed system:

In the proposed system with the use of a logistic regression algorithm that generates the best efficiency and accurate results. By the use of the stemming process which stops all root words but that the error count to be reduced and separates the label and the text so that the accuracy finding for the total data is efficient, then it converts the textual data to numerical data. The main advantage of the proposed model is it splits the training and test data, so the accuracy of the data is increased. Therefore the proposed model is implemented using the logistic regression algorithm so that the accuracy is more when compared to the existing system.

Classification Algorithm	Accuracy
Support vector Machine	0.917
Naïve Bayes Classifier	0.748
Random Forest Classifier	0.951
Decision Tree Classifier	0.78
Logistic Regression	0.89

Figure 6.1: Accuracy of Existing model

Classification Algorithm	Accuracy
Support vector Machine	0.917
Naïve Bayes Classifier	0.748
Random Forest Classifier	0.951
Decision Tree Classifier	0.78
Logistic Regression	0.98

Figure 6.2: Accuracy of Proposed model

6.3 Sample Code

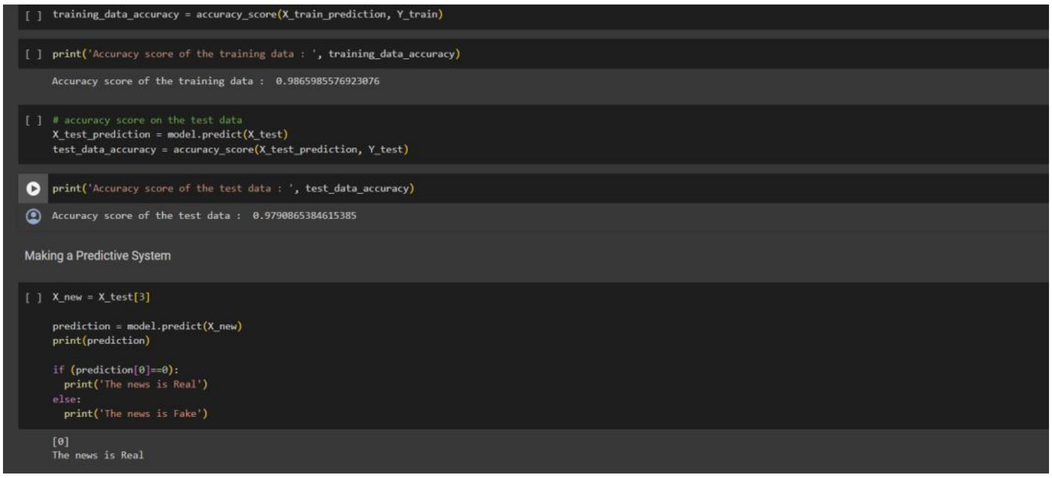
```
import numpy as np import pandas as pd
import re from nltk . corpus

import stopwords from nltk . stem . porter

import PorterStemmer from sklearn . f e a t u r e e x t r a c t i o n . t e x t import TfidfVectorizer from sklearn . model selection
import t r a i n t e s t s p l i t from sklearn . linear model import LogisticRegression from sklearn . metrics import accuracy score
import nltk nltk . download ( ' stopwords ' ) p r i n t ( stopwords . words ( ' english ' ) )
news dataset = pd . read csv ( ' / content / t r a i n . csv ' ) newsdataset . shape
news.dataset head()
news.dataset.isnull() .sum() news.dataset news.dataset. fillna('')
news.dataset 'content'] news.dataset['title']+news.dataset['author''] print(news.dataset['content'])
print (X)


print (Y)


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) printtnews.dataset content
```

#separating the data and label Xnews.dataset contentovalues news.dataset lahe values print(X)
print(Y)

Output

Figure 6.3: Graph of the model

In figure 6.3, the graph shows the accuracy of the comparison of various machine learning algorithms in which the Logistic Regression shows the best accuracy result.



```
[ ] training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[ ] print('Accuracy score of the training data : ', training_data_accuracy)

    Accuracy score of the training data :  0.9865985576923076

[ ] # accuracy score on the test data
    X_test_prediction = model.predict(X_test)
    test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

    print('Accuracy score of the test data : ', test_data_accuracy)

    Accuracy score of the test data :  0.9790865384615385

Making a Predictive System

[ ] X_new = X_test[3]

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

    if (prediction[0]==0):
      print('The news is Real')
    else:
      print('The news is Fake')

    [0]
    The news is Real
```

Figure 6.4: Accuracy of the model

In figure 6.4, it shows the accuracy of the Logistic Regression model for fake news prediction and also it predicts whether the news is fake or rea

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The main conclusion nowadays many people consume news from social media instead of traditional news media. However, social media has also been used to spread fake news, which has negative impacts on individual people and society. In that, an innovative model for fake news detection using a machine learning algorithm that is logistic regression has been presented. This model takes news events as an input and based on real-time news and classification algorithms it predicts the percentage of news being fake or real. The main conclusion of this project give you the accuracy rate of the fake news prediction with an accuracy percentage of 98%, and also there will be a clear idea of which news is real or fake.

7.2 Future Enhancements

Through the work done in this project, It shown that machine learning certainly does have the capacity to pick up on sometimes subtle language patterns that may be difficult for humans to pick up on. The next steps involved in this project come in three different aspects. The first aspect that could be improved in this project is augmenting and increasing the size of the dataset. Our feel that more data would be beneficial in ridding the model of any 25 bias based on specific patterns in the source. There is also a question as to whether or not the size of our dataset is sufficient. The second aspect in which this project could be expanded is by comparing it to humans performing the same task. Comparing the accuracies would be beneficial in deciding whether the dataset represents how difficult the task of separating fake from real news is. If humans are more accurate than the model, it may mean choosing more deceptive fake news examples. Because our acknowledge that this is only one tool in a toolbox that would be required for an end-to- end system for classifying fake news, expect that its accuracy will never reach perfect. However, it may be beneficial as a stand-alone application if its accuracy is already higher than human accuracy at the same task. In addition to comparing the accuracy to human accuracy, it would also be interesting to compare the phrases/trigrams that a human would point out if asked what they based their classification decision on. It could quantify how similar these patterns are to those that humans find indicative of fake and real news. Finally, our mentioned throughout, this application is the only one that would be necessary for a larger toolbox that could function as a highly accurate fake news classifier. Other tools that would need to be built may include a fact detector and a stance detector. To combine all of these “routines,” there would need to be some type of model that combines all of the tools and learns how to weigh each of them in its final decision.

CHAPTER 8 SOURCE CODE

8.1 Source Code

```
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Embedding, SpatialDropout1D, Conv1D, MaxPooling1D, Dropout
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report

# Load dataset
data = pd.read_csv('Fake.csv')  # Replace with your dataset path

# Check columns in the dataset
print(data.columns)

# Combine title and text into a new column 'content' for processing
data['content'] = data['title'] + ' ' + data['text']

# Check if 'label' exists or create a dummy label for testing
if 'label' not in data.columns:
    # Example: Creating a dummy label for illustration
    # In real scenarios, ensure you have actual labels
    data['label'] = np.random.choice([0, 1], size=len(data))

```

```
# Preprocess data
X = data['content']  # Column containing combined title and text
y = data['label']    # Column containing labels (0 for real, 1 for fake)
# Encode labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
# Tokenization
max_words = 5000  # Maximum number of words
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(X)
X_sequences = tokenizer.texts_to_sequences(X)
# Padding sequences
max_len = 100  # Maximum length of each sequence
X_padded = pad_sequences(X_sequences, maxlen=max_len)
# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X_padded, y, test_size=0.2, random_state=42)

# Model definition
model = Sequential()
model.add(Embedding(max_words, 128, input_length=max_len))
model.add(SpatialDropout1D(0.2))

# CNN Layer
model.add(Conv1D(filters=64, kernel_size=5, activation='relu'))
```

```
model.add(MaxPooling1D(pool_size=2))

# LSTM Layer
model.add(LSTM(100, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(100))
model.add(Dropout(0.2))

# Fully connected layer
model.add(Dense(1, activation='sigmoid')) # Binary classification
# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Model Training
batch_size = 64
epochs = 5
model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, validation_data=(X_test, y_test))

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print(f'Test Accuracy: {accuracy:.2f}')
# Predictions and Classification Report
y_pred = (model.predict(X_test) > 0.5).astype("int32")
print(classification_report(y_test, y_pred))
```

CHAPTER 9 REFERENCES

- Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, Anupam Joshi (2014),Analysing and Measuring the Spread of Fake Content on Twitter during High Impact Events". Security and Privacy Symposium , CSE - IIT-Kanpur.
- Kai Shu, Deepak Mahudeswaran, Huan Liu (2019) , FakeNewsTracker: a tool for fake news collection, detection, and visualization,Computational Mathematical Organization Theory, vol. 25 issue 1, pp. 60-71.
- Kai Shu , Suhang Wang, Huan Liu,(2018), Understanding UserProfiles on Social Media for Fake News Detection",IEEE Conference..
- Minjung Park and Sangmi Cha (2023).Constructing a User-Centered Fake News Detection Model by Using Classification Algorithms in Machine Learning Techniques, IEEE Journal paper.
- Mykhailo Granik, Volodymyr Mesyura,(2018),"Fake News Detection Using Naive Bayes Classifier", IEEE First Ukraine Conference on Electrical and Computer Engineering, IEEE, pp. 900–903.
- Marco L Della Vedova et al.(2018), Automatic online fake news detection combining content and social signals. In 22nd Conference of Open Innovations Association (FRUCT). IEEE conference paper, pp. 272–279.
- R. A. Monteiro, R. L. Santos, T. A. Pardo, T. A. d. Almeida, E. E. Ruiz, and O. A. Vale (2018),Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results", 13th International Conference.
- Raveena Dayani, Nikita Chhabra, Taruna Kadian and Rishabh Kaushal,(2015), Rumor detection in twitter: An analysis in retrospect", IEEE Conference.
- Sardar Hamidian and Mona Diab. (2015),Rumor Detection and Classification for Twitter Data". SOTICS 2015: The Fifth International Conference on Social Media Technologies, Communication, and Informatics,IEEE conference.
- Van Der Walt, Estee, and Jan Eloff.(2018), Using machine learning to detect´ fake identities: bots vs humans, IEEE Conference , 6540-6549