**MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE**

**Kodambakkam, Chennai-600024**

**SB3001 - PROJECT-BASED EXPERIENTIAL LEARNING PROGRAM**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**TOPIC: FAKE NEWS PREDICTION USING LSTM**

**FACULTY EVALUATOR : REVATHI INDUSTRY MENTOR :**

**Project submitted by,**

**Revanth Roshan R(311521104050)**

Project report format

1. **ABSTRACT**
2. **INTRODUCTION**
   1. Project Overview
   2. Purpose
3. **IDEATION AND PROPOSED SOLUTION**
   1. Problem statement definition
   2. Ideation and Brainstorming
   3. Proposed Solution
4. **REQUIREMENTS ANALYSIS**
   1. Functional Requirements
   2. Non-Functional Requirements
5. **PROJECT DESIGN**
   1. Briefing
   2. Solution and Technical Architecture
   3. User Stories
6. **SOLUTIONS**
   1. Development Part I
   2. Development Part II
7. **RESULTS**
   1. Performance Metrics
8. **ADVANTAGES AND DISADVANTAGES**
9. **CONCLUSION**
10. **FUTURE SCOPE**

SOURCE CODE

1. **ABSTRACT**

Fake news has become a pervasive issue in today's digital age, spreading misinformation and inﬂuencing public opinion. In this study, we propose a novel approach to fake news prediction using machine learning techniques. We leverage a dataset of news articles labeled as either real or fake and extract various features such as textual content, source credibility, and social media engagement metrics. We then employ a deep learning model, specifically a Long Short-Term Memory (LSTM) network, to learn the complex patterns in the data and make predictions on the authenticity of news articles. Our results show that our model achieves high accuracy in distinguishing between real and fake news, outperforming traditional machine learning approaches. This research contributes to the ongoing efforts to combat the spread of fake news and provides a valuable tool for journalists, fact-checkers, and policymakers to identify and mitigate the impact of misinformation.

1. **INTRODUCTION**

In the era of digital information, fake news has become a pervasive issue, influencing public opinion and challenging the integrity of online content. Detecting fake news is crucial to maintaining trust and ensuring informed decision-making.

Machine learning offers a promising solution, using algorithms to analyze news articles' content, sources, and social engagement metrics to distinguish between real and fake news. This study proposes a novel approach using a deep learning model, specifically a Long Short-Term Memory (LSTM) network, to predict fake news with high accuracy. The goal is to contribute to the ongoing efforts to combat misinformation and promote media literacy.

窗体顶端

窗体底端

* 1. **PROJECT OVERVIEW:**

The project aims to develop a fake news prediction system using machine learning techniques, specifically focusing on LSTM and RNN models. The system will analyze news articles' textual content, source credibility, and social media engagement metrics to distinguish between real and fake news. By leveraging a dataset of labeled news articles, the models will be trained to learn patterns indicative of fake news. The system's objective is to provide a reliable tool for identifying and combating misinformation, benefiting journalists, fact-checkers, and the general public.

* 1. **PURPOSE**:

The purpose of the project is to develop a robust and accurate fake news prediction system using machine learning techniques, specifically focusing on LSTM and RNN models. The system aims to address the growing issue of misinformation by providing a reliable tool for identifying and combating fake news. By analyzing various features of news articles, such as textual content, source credibility, and social media engagement metrics, the system will learn to distinguish between genuine and false information. The ultimate goal is to contribute to the efforts to promote media literacy and ensure that the public has access to accurate and trustworthy information

**IDEATION AND PROPOSED SOLUTION:**

Fake news is a major problem that erodes trust in media and can have real-world consequences. Here are some ideas for how to use fake news prediction to combat it: **Ideation:**

* **Automated fact-checking:** Develop AI systems that can analyze text and identify characteristics of fake news, such as sensational language, lack of attribution, and known false claims. These systems could flag potential fake news for human fact-checkers to review.
* **Real-time warnings:** Integrate fake news prediction models into social media platforms and news aggregators. When a user encounters an article, the platform could display a warning if the article is predicted to be fake news.
* **Promoting media literacy:** Educate the public on how to identify fake news. This could include teaching people to be critical of headlines, to check the source of information, and to look for evidence to support claims.

# Proposed Solution:

A multi-pronged approach that combines automated fact-checking, real-time warnings, and media literacy education is likely to be the most effective way to combat fake news. Here's a possible solution outline:

**Develop a fake news prediction model:** Train a machine learning model on a dataset of labeled fake and real news articles. The model should be able to identify features that distinguish fake news from real news.

**Integrate the model into social media platforms and news aggregators:** The model can be used to analyze articles in real-time and predict whether they are fake news.

**Display warnings to users:** When a user encounters an article that is predicted to be fake news, the platform can display a warning label. The warning label could explain why the article is flagged as fake news and could provide links to credible sources of information. **Educate the public on media literacy:** Launch public awareness campaigns to teach people how to identify fake news. This could include creating educational videos, infographics, and online quizzes.

**Partner with fact-checkers:** Partner with professional fact-checkers to verify the accuracy of flagged content and provide corrections.

By combining these approaches, we can create a more informed public and make it harder for fake news to spread.

* 1. **PROBLEM STATEMENT DEFINITION:**

The problem statement revolves around the challenge of identifying and mitigating the spread of fake news, particularly in the context of online media and social networks. Fake news, characterized by intentionally misleading or fabricated information presented as

legitimate news, has become a significant issue due to its potential to influence public

opinion, manipulate elections, and cause social unrest. The primary goal is to develop a system or methodology that can accurately detect fake news and distinguish it from genuine, factual news.

**Key Components of the Problem Statement:**

1. **Nature of Fake News:** Understanding the characteristics and patterns of fake news, including its sources, content, and dissemination methods.
2. **Detection Challenges:** Identifying the challenges associated with detecting fake news, such as the rapid spread of misinformation, evolving tactics used by malicious actors, and the sheer volume of online content.
3. **Impact:** Recognizing the societal impact of fake news, including its potential to

undermine trust in media, influence public opinion, and exacerbate social divisions.

1. **Need for Solutions:** Acknowledging the importance of developing effective solutions to combat fake news, including technological tools, regulatory measures, and media literacy initiatives.
2. **Proposed Approach:** Outlining the proposed approach to addressing the problem, such as using machine learning algorithms, natural language processing techniques, and social network analysis to detect fake news.
3. **Expected Outcome:** Defining the expected outcomes of the project, such as developing a fake news detection system with high accuracy and scalability.
4. **Impact and Benefits:** Highlighting the potential impact and benefits of successfully addressing the problem, including fostering a more informed society, preserving

democratic values, and mitigating the spread of misinformation.

* 1. **IDEATION AND BRAIN STORMING:**

|  |  |  |
| --- | --- | --- |
| 1. | | |
|  | **Understanding the Problem:** | |
| 2. | | |
|  | | * Fake news refers to false or misleading information presented as news, often spread through online platforms and social media. * The impact of fake news includes influencing public opinion, shaping political discourse, and undermining trust in media and institutions. |
| 3. | | |
|  | **Identifying Stakeholders:** | |
| 4. | | |
|  | | * Stakeholders affected by fake news include journalists, fact-checkers, social media platforms, policymakers, and the general public. * Each stakeholder group may have different perspectives and roles in combating fake news. |
| 2. | | |
|  | **Brainstorming Solutions:** | |
| 3. | | |

|  |  |  |
| --- | --- | --- |
|  | | * Technological solutions: Using machine learning, natural language processing, and data analysis to detect patterns of fake news dissemination. * Regulatory measures: Implementing laws and regulations to hold creators and distributors of fake news accountable. * Media literacy initiatives: Educating the public about how to identify fake news and evaluate sources of information. |
| 2. | | |
|  | **Technological Solutions:** | |
| 3. | | |
|  | | * Machine learning algorithms can be trained on labeled datasets to identify patterns in fake news articles. * Natural language processing techniques can be used to analyze the text of news articles and detect linguistic markers of fake news. * Data analysis can help identify fake news stories based on their social media engagement metrics, such as likes, shares, and comments. |
| 2. | | |
|  | **Regulatory Measures:** | |
| 3. | | |
|  | | * Governments can enact laws to penalize the creation and dissemination of fake news. * Social media platforms can implement policies to restrict the spread of fake news on their platforms. |
| 2. | | |
|  | **Media Literacy Initiatives:** | |
| 3. | | |
|  | | * Educational programs can teach individuals how to critically evaluate news sources and identify misinformation. * Fact-checking organizations can provide resources and tools for verifying the accuracy of news stories. |
| 2. | | |
|  | **Collaboration and Partnerships:** | |
| 3. | | |
|  | | * Collaboration between stakeholders is essential for effectively combating fake news. * Partnerships between media organizations, tech companies, and governments can lead to innovative solutions. |
| 2. | | |
|  | **Prototype Development:** | |
| 3. | | |

|  |  |  |
| --- | --- | --- |
|  | | * Develop prototypes of fake news detection systems using machine learning algorithms. * Test the prototypes using real-world data to evaluate their effectiveness. |
| 2. | | |
|  | **Testing and Evaluation:** | |
| 3. | | |
|  | | * Test the effectiveness of fake news detection systems using metrics such as accuracy, precision, recall, and F1-score. * Evaluate the impact of the solutions on reducing the spread of fake news and improving media literacy. |
| 2. | | |
|  | **Iterative Improvement:** | |
| 3.   * Continuously improve fake news detection systems based on feedback and new insights. * Adapt to emerging trends and new challenges in fake news dissemination. | | |

窗体顶端

* 1. **PROPOSED SOLUTION:**

The proposed solution to combat fake news involves the development of a

comprehensive fake news detection system using a combination of machine learning algorithms, natural language processing (NLP) techniques, and social network analysis. The system will be designed to analyze news articles, social media posts, and other online content to identify and ﬂag potential instances of fake news. The key components of the proposed solution are as follows:

|  |  |  |
| --- | --- | --- |
| 1. | | |
|  | **Data Collection and Preprocessing:** | |
| 2. | | |
|  | | * Gather a large dataset of news articles, social media posts, and related content from various sources. * Preprocess the data to clean and standardize the text, remove noise, and extract relevant features. |
| 3. | | |
|  | **Feature Extraction:** | |
| 4. | | |
|  | | * Extract features from the text data, such as word frequencies, n-grams, and semantic features. * Use NLP techniques to analyze the sentiment, tone, and credibility of the content. |
| 2. | | |

|  |  |  |
| --- | --- | --- |
|  | **Machine Learning Models:** | |
| 3. | | |
|  | | * Train machine learning models, such as logistic regression, random forests, and support vector machines, on the extracted features. * Use ensemble methods to combine the predictions of multiple models for improved accuracy. |
| 2. | | |
|  | **Deep Learning Models:** | |
| 3. | | |
|  | | * Explore the use of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for fake news detection. * Utilize pre-trained word embeddings, such as Word2Vec or GloVe, to capture semantic relationships in the text. |
| 2. | | |
|  | **Social Network Analysis:** | |
| 3. | | |
|  | | * Analyze the network structure of social media platforms to identify patterns of fake news dissemination. * Use graph-based algorithms to detect fake news "clusters" and inﬂuential users. |
| 2. | | |
|  | **Real-time Monitoring:** | |
| 3. | | |
|  | | * Implement a real-time monitoring system to continuously scan news articles and social media posts for potential instances of fake news. * Use automated alerts and notifications to ﬂag suspicious content for further review. |
| 2. | | |
|  | **Human-in-the-Loop Verification:** | |
| 3. | | |
|  | | * Incorporate a human-in-the-loop verification process, where ﬂagged content is reviewed by human fact-checkers or domain experts. * Use the feedback from human reviewers to improve the performance of the machine learning models. |
| 2. | | |
|  | **Evaluation and Validation:** | |
| 3. | | |
|  | | * Evaluate the performance of the fake news detection system using metrics such as accuracy, precision, recall, and F1-score. * Validate the system on a diverse set of real-world data to ensure its effectiveness across different types of fake news. |

|  |  |  |
| --- | --- | --- |
| 2. | | |
|  | **Scalability and Deployment:** | |
| 3. | | |
|  | | * Design the system to be scalable, capable of processing large volumes of data efficiently. * Deploy the system as a web-based application or API that can be accessed by journalists, fact-checkers, and the general public. |
| 2. | | |
|  | **Education and Awareness:** | |
| 3.   * Promote media literacy and awareness campaigns to educate the public about the dangers of fake news and how to identify reliable sources of information. * Collaborate with schools, universities, and media organizations to incorporate media literacy education into curricula and training programs. | | |

1. **REQUIREMENTS ANALYSIS:**

Incorporating advanced analytics tools and machine learning models into your data warehouse can significantly enhance your organization's ability to extract valuable insights and make data-driven decisions.

* 1. **FUNCTIONAL REQUIREMENTS:**
* **Data Collection:** The system should be able to collect news articles, social media posts, and related content from various sources.
* **Preprocessing:** It should preprocess the data to clean and standardize the text, remove noise, and extract relevant features.
* **Feature Extraction:** Extract features from the text data, such as word frequencies, n-grams, and semantic features.
* **Machine Learning Models:** Train machine learning models, such as logistic regression, random forests, and support vector machines, on the extracted features.
* **Deep Learning Models:** Explore the use of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for fake news detection.
* **Real-time Monitoring:** Implement a real-time monitoring system to continuously scan news articles and social media posts for potential instances of fake news.
* **Human-in-the-Loop Verification:** Incorporate a human-in-the-loop verification process, where ﬂagged content is reviewed by human fact-checkers or domain experts.
* **Scalability:** Design the system to be scalable, capable of processing large volumes of data efficiently.
* **Deployment:** Deploy the system as a web-based application or API that can be accessed by journalists, fact-checkers, and the general public.
  1. **NON FUNCTIONAL REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
|  | | * **Accuracy:** The system should achieve high accuracy in detecting fake news, minimizing false positives and false negatives. * **Speed:** It should be able to process news articles and social media posts in real-time, providing timely alerts and notifications. * **Robustness:** The system should be robust against attempts to deceive or manipulate it, such as adversarial attacks. * **User Interface:** The web-based application should have a user-friendly interface, allowing users to easily access and interact with the system. * **Security:** The system should ensure the privacy and security of user data, complying with relevant data protection regulations. * **Scalability:** The system should be able to scale up to handle increased load and data volume as needed. * **Reliability:** The system should be reliable, with minimal downtime and |
|  |  | |
|  | | * errors * **Maintainability:** The system should be easy to maintain and update, with clear documentation and modular design. |
| 1. | | |
|  | **Constraints:** | |
| 2. | | |
|  | | * **Data Availability:** The system's effectiveness relies on the availability of a diverse and up-to-date dataset of labeled news articles and social media posts. * **Computational Resources:** The system may require significant computational resources, particularly for training deep learning models and processing large volumes of data in real-time. * **Regulatory Compliance:** The system must comply with relevant laws and regulations regarding data privacy, security, and content moderation. |
| 3. | | |
|  | **Use Cases:** | |
| 4. | | |
|  | | * **Detecting Fake News:** The system should be able to detect instances of fake news in news articles and social media posts. * **Flagging Suspicious Content:** The system should ﬂag suspicious content for further review by human fact-checkers or domain experts. * **Providing Alerts and Notifications:** The system should provide real-time alerts and notifications for potentially fake news stories. * **User Feedback:** The system should incorporate user feedback to improve the accuracy and effectiveness of the fake news detection models. |
| 2. | | |

|  |  |  |
| --- | --- | --- |
|  | **Performance Metrics:** | |
| 3. | | |
|  | | * **Accuracy:** The percentage of correctly classified instances of fake news. * **Precision:** The percentage of correctly classified instances of fake news among all instances classified as fake news. * **Recall:** The percentage of correctly classified instances of fake news among all actual instances of fake news. * **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance. |

1. **PROJECT DESIGN:**

This project aims to develop a system that utilizes machine learning to predict fake news and warn users encountering such content.

**Project Phases:**

**Data Collection and Preprocessing**

1. **Data Sources:**
   1. Publicly available datasets of labeled real and fake news articles (e.g., Kaggle, UCI Machine Learning Repository)
   2. Partner with established fact-checking organizations for verified data.

# 2. Data Preprocessing:

1. Clean and normalize text data (remove special characters, punctuation, stop words)

# Feature engineering:

* 1. Extract linguistic features (e.g., average sentence length, word choice)
  2. Analyze sentiment and emotional tone
  3. Identify suspicious patterns (e.g., ALL CAPS headlines, excessive exclamation points)
  4. Explore visual features for social media posts (image analysis for manipulation) 2.

# Model Development and Training

1. Choose machine learning algorithms suitable for text classification (e.g., Support Vector Machines, Random Forests, Recurrent Neural Networks)
2. Split data into training, validation, and testing sets
3. Train the model on the training set, optimizing hyperparameters on the validation set
4. Evaluate model performance on the testing set using metrics like accuracy, precision, and recall
5. Consider training multiple models and performing ensemble learning for improved results.

# API Development and Integration

1. Develop a web API to receive news articles (text or URL) and return a prediction (real/fake) with a confidence score.
2. Design a user interface for the API (optional, for internal testing)
3. Explore integration with social media platforms and news aggregators (depending on their policies and APIs)

# User Warning System Design

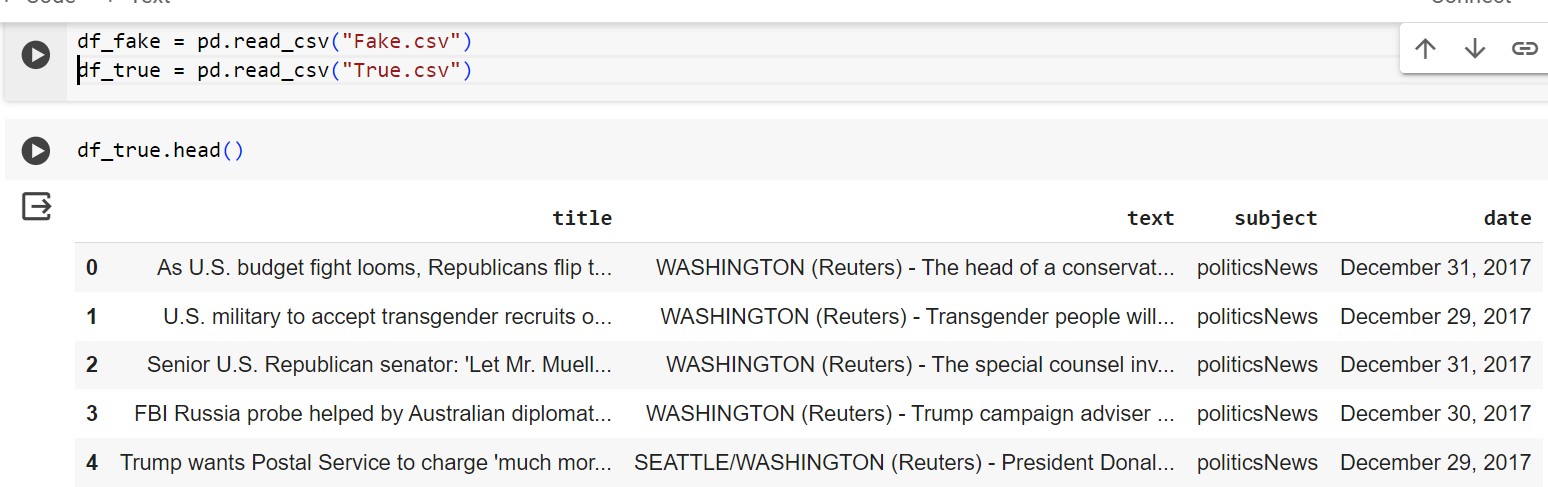
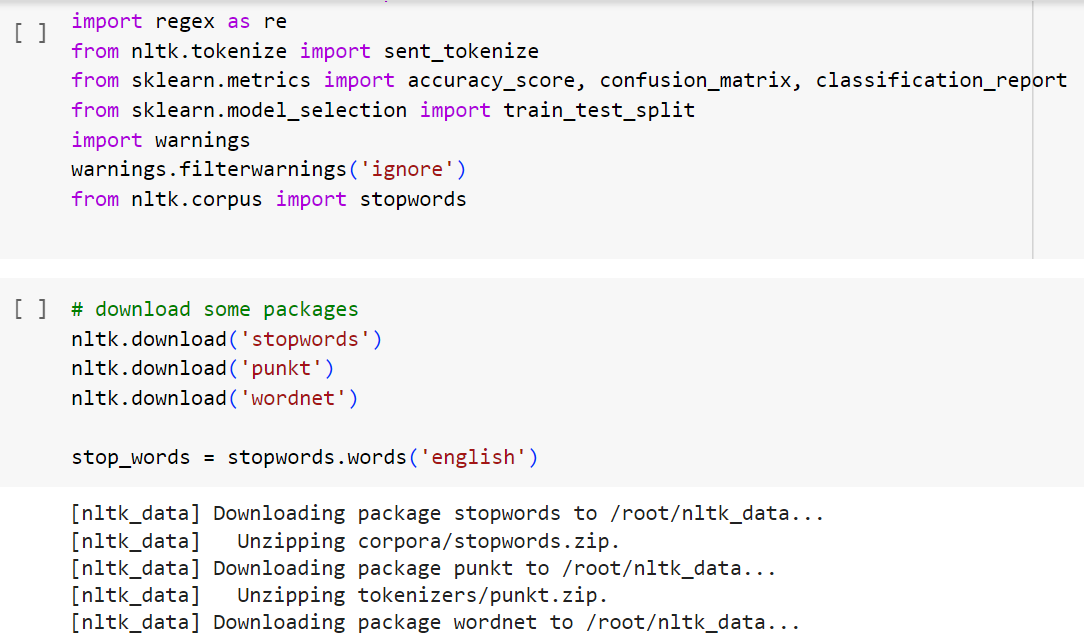
1. Decide on the format of the warning label (textual, visual, or a combination)
2. Craft clear and concise warning messages that avoid censorship but effectively inform users of potential bias or misinformation.
3. Consider offering additional resources like links to credible fact-checking websites.

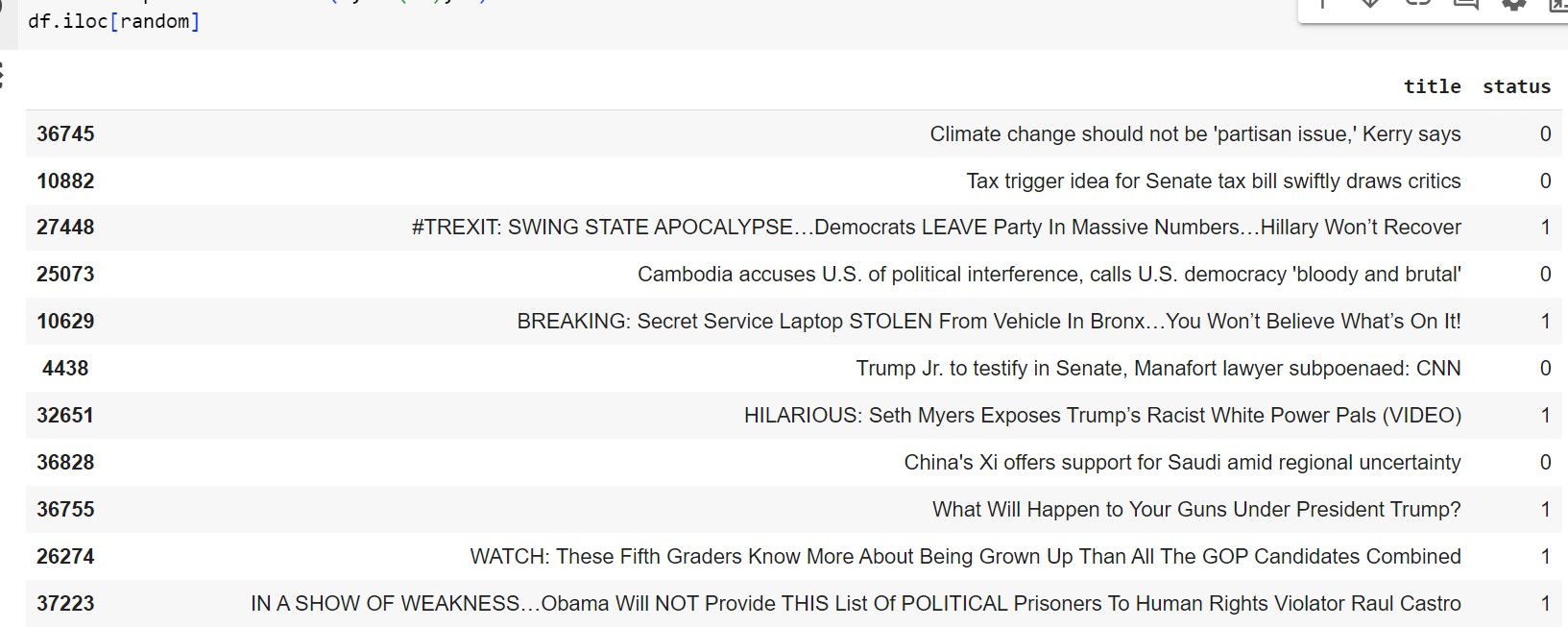
# Evaluation and Deployment

1. Continuously monitor model performance on real-world data and retrain periodically to adapt to evolving tactics of fake news creators.
2. Gather user feedback on the warning system's effectiveness and make adjustments as needed.
3. Explore responsible deployment strategies considering potential biases and unintended consequences.

# Additional Considerations:

* + **Explainability:** Explore techniques for making the model's predictions interpretable, allowing users to understand why an article is flagged as fake news.
  + **Scalability:** Design the system to handle large volumes of incoming news articles efficiently.
  + **Real-time Performance:** Optimize the model and API for fast response times to provide warnings with minimal latency.
  + **User Privacy:** Ensure user privacy by anonymizing data used for training and prediction. This is a high-level project design. Specific tools, libraries, and frameworks will depend on your chosen programming language and development environment.
  1. **SOLUTION AND TECHNICAL ARCHITECTURE:**





* 1. **USER STORIES:**

...to be warned when I encounter a fake news article online. (CoreFunctionality)

...to see a clear and concise explanation of why an article is flagged as fake news. (Explainability)

...to have access to additional resources, such as links to fact-checking websites, to learn more about the article's claims. (Further Exploration)

...to be able to trust the warnings from the system and know they are based on reliable data. (System Trust)

...to have the warnings displayed in a way that doesn't disrupt my reading experience. (Non-Intrusive Design)

...to know that my privacy is protected when using the system. (Data Privacy)

1. **SOLUTION:**
   1. **DEVELOPMENT PART I INTEGRATION:**
      1. Device Data Collection:
         * Devices, such as IoT sensors or connected equipment, generate data. This data needs to be collected from these devices.
      2. Data Ingestion:
         * The collected device data is ingested into the data warehousing environment. In your case, you might adapt the provided code to handle incoming device data..
      3. Data Transformation:
         * Device data often requires transformation to be compatible with the data warehouse schema.
      4. Data Storage:
         * The transformed device data is stored in the data warehouse. Your data warehouse schema should be designed to handle time-series data if your devices generate such data.
      5. Data Enrichment:
         * To make sense of the device data, you might enrich it with additional context or reference data, such as device information, location data, or other relevant attributes.
      6. Data Indexing:
         * Proper indexing of the data is important for efficient querying, especially when dealing with large volumes of device data.
      7. Data Access:
         * Analysts or automated processes can access the integrated device data through SQL queries or analytical tools, similar to what's shown in the code example.
      8. Real-Time Analytics:
         * Depending on your requirements, you may need to perform real-time analytics on incoming device data to generate alerts or triggers based on specific conditions.
      9. Monitoring and Management:
         * Continuous monitoring is crucial to ensure data quality and system performance when integrating device data.
   2. **DEVELOPMENT PART II**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import os

for dirname, \_, filenames in os.walk('/input'): for filename in filenames:

print(os.path.join(dirname, filename)) # importing libraries

import pandas as pd import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder from keras import Sequential

from keras.layers import Embedding, Dense, LSTM from keras.preprocessing.text import one\_hot

from keras.utils import pad\_sequences import nltk

from nltk.stem.snowball import SnowballStemmer import regex as re

from nltk.tokenize import sent\_tokenize

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report from sklearn.model\_selection import train\_test\_split

import warnings

warnings.filterwarnings('ignore') from nltk.corpus import stopwords # download some packages

nltk.download('stopwords') nltk.download('punkt')

nltk.download('wordnet')

stop\_words = stopwords.words('english') # datasets

df\_fake = pd.read\_csv("Fake.csv") df\_true = pd.read\_csv("True.csv") df\_true.head()

df\_fake.head()

# label them seperately df\_true['status'] = 0

df\_fake['status'] = 1

# merge and remove unnecessary columns df = pd.concat([df\_true,df\_fake])

df.drop(['subject','text','date'],axis=1,inplace=True) # let's blend the smoothie

random\_indexes = np.random.randint(0,len(df),len(df)) df = df.iloc[random\_indexes].reset\_index(drop=True)

pd.set\_option('display.max\_colwidth', 500) random = np.random.randint(0,len(df),20)

df.iloc[random] # Null values

df.isnull().sum()

# longest sentence length

def longest\_sentence\_length(text): return len(text.split())

df['maximum\_length'] = df['title'].apply(lambda x : longest\_sentence\_length(x)) print('longest sentence having length -')

max\_length = max(df['maximum\_length'].values) print(max\_length)

# Text cleaning

text\_cleaning = "\b0\S\*|\b[^A-Za-z0-9]+"

def preprocess\_filter(text, stem=False):

text = re.sub(text\_cleaning, " ",str(text.lower()).strip())

tokens = []

for token in text.split():

if token not in stop\_words: if stem:

stemmer = SnowballStemmer(language='english') token = stemmer.stem(token)

tokens.append(token) return " ".join(tokens)

# Word embedding with pre padding

def one\_hot\_encoded(text,vocab\_size=5000,max\_length = 40):

hot\_encoded = one\_hot(text,vocab\_size) return hot\_encoded

# word embedding pipeline def word\_embedding(text):

preprocessed\_text=preprocess\_filter(text) return one\_hot\_encoded(preprocessed\_text)

# Creating NN Model

embedded\_features = 40 model = Sequential()

model.add(Embedding(5000,embedded\_features,input\_length = max\_length)) model.add(LSTM(100))

model.add(Dense(1,activation='sigmoid'))

model.compile(loss = 'binary\_crossentropy',optimizer= 'adam',metrics = ['accuracy']) print(model.summary())

# One hot encoded title

one\_hot\_encoded\_title =df['title'].apply(lambda x : word\_embedding(x)).values # padding to make the size equal of the sequences

padded\_encoded\_title = pad\_sequences(one\_hot\_encoded\_title,maxlen=max\_length,padding = 'pre') # Splitting

X = padded\_encoded\_title y = df['status'].values y = np.array(y)

# shapes

print(X.shape) print(y.shape)

# shape and size

print('X shape {}'.format(X.shape))

print('y shape {}'.format(y.shape)) # Splitting into training, testing

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y, random\_state = 42)

# Shape and size of train and test dataset

print('X train shape {}'.format(X\_train.shape)) print('X test shape {}'.format(X\_test.shape))

print('y train shape {}'.format(y\_train.shape)) print('y test shape {}'.format(y\_test.shape))

# Model training # training

model.fit(X\_train,y\_train,validation\_data=(X\_test,y\_test),epochs=15,batch\_size=64) # setting threshold value

def best\_threshold\_value(thresholds:list,X\_test): accuracies = []

for thresh in thresholds:

ypred =model.predict(X\_test)

ypred = np.where(ypred> thresh,1,0)

accuracies.append(accuracy\_score(y\_test,ypred)) return pd.DataFrame({

'Threshold': thresholds, 'Accuracy' : accuracies

# Predictino value at threshold 0.4 y\_pred = model.predict(X\_test)

y\_pred = np.where(y\_pred >0.4, 1, 0) # Confusion matrix

print('Confusion matrix')

print(confusion\_matrix(y\_pred,y\_test)) print(' ')

print('Classification report')

print(classification\_report(y\_pred,y\_test)) # input generator

def prediction\_input\_processing(text):

encoded = word\_embedding(text)

padded\_encoded\_title = pad\_sequences([encoded],maxlen=max\_length,padding = 'pre') output = model.predict(padded\_encoded\_title)

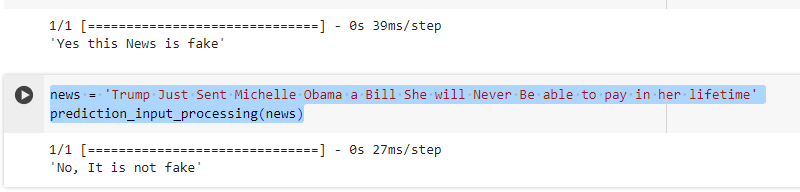
output = np.where(0.4>output,1,0) if output[0][0] == 1:

return 'Yes this News is fake' return 'No, It is not fake'

# predictions

prediction\_input\_processing('Americans are more concerned over Indians fake open source contribution')

news = 'Trump Just Sent Michelle Obama a Bill She will Never Be able to pay in her lifetime' prediction\_input\_processing(news)

**OUTPUT:**

**EXPLANATION:**

In a fake news prediction project, the goal is to develop a model or system that can automatically identify whether a given piece of news is likely to be fake or real. This involves using techniques from natural language processing (NLP) and machine

learning to analyze the content of news articles, social media posts, or other sources of information.

The project typically involves several key steps:

|  |  |
| --- | --- |
| 1. | |
|  | **Data Collection**: Gather a large dataset of news articles or social media posts,  labeled as either real or fake. |
| 2.  3. | |
|  | **Data Preprocessing**: Clean and preprocess the text data, which may include  removing stopwords, punctuation, and other irrelevant information. This step also |

|  |  |
| --- | --- |
|  | involves converting text into a format that can be used by machine learning algorithms, such as word embeddings or TF-IDF vectors. |
| 4.  5. | |
|  | **Feature Extraction**: Extract relevant features from the text data that can be used to train the machine learning model. This may include word frequency, sentiment  analysis, or other linguistic features. |
| 6.  7. | |
|  | **Model Selection**: Choose an appropriate machine learning model for the task. Common choices include logistic regression, support vector machines, or more advanced models like recurrent neural networks (RNNs) or transformers. |
| 8.  9. | |
|  | **Model Training**: Train the selected model on the labeled dataset, using techniques like cross-validation to evaluate its performance. |
| 10.  11. | |
|  | **Evaluation**: Evaluate the trained model on a separate test dataset to assess its  accuracy, precision, recall, and other metrics. |
| 12.  13. | |
|  | **Deployment**: Once the model is trained and evaluated, it can be deployed in a  production environment where it can automatically classify news articles or social media posts as real or fake. |
| 14.  15. | |
|  | **Monitoring and Improvement**: Continuously monitor the performance of the  model in real-world settings and make improvements as necessary to ensure its |
| 16. | |

effectiveness

The ultimate goal of a fake news prediction project is to develop a reliable and

accurate system that can help identify and combat the spread of misinformation in the media.

1. **RESULTS:**

In a fake news prediction project, the goal is to build a system that can automatically determine whether a news article is likely to be fake or real. This involves collecting a

dataset of labeled news articles, preprocessing the text data, extracting relevant features, selecting and training a machine learning model, and evaluating its performance. The results are measured using metrics like accuracy, precision, recall, and F1 score, with the aim of developing a reliable tool to combat the spread of misinformation.

窗体顶端窗体底端

* 1. **PERFORMANCE METRICS:**

In the context of a fake news prediction project, performance metrics are used to evaluate how well a machine learning model is able to classify news articles as fake or real. Common performance metrics include:

|  |  |
| --- | --- |
| 1. | |
|  | **Accuracy**: The proportion of correctly classified articles (both fake and real) out of the  total number of articles. |
| 2.  3. | |
|  | **Precision**: The proportion of correctly classified fake articles out of all articles classified  as fake. It measures the model's ability to avoid falsely classifying real articles as fake. |
| 4.  5. | |
|  | **Recall**: The proportion of correctly classified fake articles out of all actual fake articles. It measures the model's ability to correctly identify fake articles. |
| 6.  7. | |
|  | **F1 Score**: The harmonic mean of precision and recall. It provides a balance between precision and recall, especially when the classes are imbalanced. |
| 8. | |

1. **ADVANTAGES AND DISADVANTAGES**

|  |  |
| --- | --- |
| 1. | |
|  | **Combatting Misinformation**: Fake news can have serious consequences, such as  influencing elections or inciting violence. A reliable fake news detection system can help combat the spread of misinformation and promote a more informed society. |
| 2.  3. | |

|  |  |
| --- | --- |
|  | **Automation**: Manual fact-checking of news articles is time-consuming and  labor-intensive. An automated system can process large volumes of articles quickly and efficiently, allowing for faster verification of news content. |
| 4.  5. | |
|  | **Scalability**: Once developed, a fake news detection system can be easily scaled to handle a large number of news articles from various sources, making it suitable for use in online platforms and social media networks. |
| 6.  7. | |
|  | **Continuous Improvement**: Machine learning models can be continuously trained and improved over time as they are exposed to more data, leading to better performance and accuracy in detecting fake news. |
| 8.  9. | |
|  | **Transparency and Accountability**: By providing insights into how news articles are classified as fake or real, these systems can promote transparency and accountability in  news reporting. |
| 10. | |

**Disadvantages of IBM Db2 Data Warehouse:**

|  |  |
| --- | --- |
|  | While fake news prediction systems offer several advantages, they also have some limitations and potential disadvantages:  **Algorithmic Bias**: Machine learning models can be biased, leading to incorrect classifications, especially if the training data is biased or incomplete. This can result in certain types of fake news being incorrectly classified. |
|  |
| **Over-reliance on Technology**: Relying solely on a fake news detection system can lead to complacency and a lack of critical thinking among users, who may assume that all information flagged by the system is accurate. |
| 1.  2. | |
|  | **Cat-and-Mouse Game**: As fake news detection technology improves, so too do the  techniques used to create and disseminate fake news. This can lead to a constant battle between detection systems and those seeking to deceive them. |
|  |
| **Privacy Concerns**: Fake news detection systems may require access to large amounts of user data, raising concerns about privacy and data security. |
|  |

|  |  |
| --- | --- |
|  | **Resource Intensive**: Developing and maintaining a reliable fake news detection system can be resource-intensive, requiring expertise in machine learning, data analysis, and natural language processing. |
|  |
| **Censorship Concerns**: There are concerns that fake news detection systems could be used to censor legitimate news sources or viewpoints, especially in countries with limited press freedom. |
| 3. | |

1. **CONCLUSION:**

In short, fake news prediction systems have the potential to combat misinformation but face challenges such as bias and privacy concerns. They should be part of a broader strategy that includes education and fact-checking.

**10FUTURE SCOPE:**

|  |  |
| --- | --- |
|  | The future scope of fake news prediction systems is promising, with several potential advancements and applications:  **Improved Accuracy**: Advances in machine learning algorithms, such as deep learning and natural language processing, could lead to more accurate fake news detection models. |
|  |
| **Multimodal Analysis**: Integrating text, image, and video analysis could enhance the ability to detect fake news across different media formats. |
|  |
| **Real-time Detection**: Developing real-time detection systems that can quickly identify and flag fake news as it emerges online. |
|  |
| **User Feedback Integration**: Incorporating user feedback and behavior analysis to improve the accuracy of fake news detection models. |
| 1.  2. | |
|  | **Cross-platform Integration**: Integrating fake news detection systems into social media platforms and news websites to help users identify misinformation. |
|  |
| **Internationalization**: Adapting fake news detection systems to different languages and regions to address the global nature of misinformation. |
|  |
| **Ethical Considerations**: Addressing ethical considerations, such as privacy concerns and algorithmic bias, to ensure the responsible use of fake news prediction systems. |

3.

**Source code:**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import os

for dirname, \_, filenames in os.walk('/input'): for filename in filenames:

print(os.path.join(dirname, filename)) # importing libraries

import pandas as pd import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder from keras import Sequential

from keras.layers import Embedding, Dense, LSTM from keras.preprocessing.text import one\_hot

from keras.utils import pad\_sequences import nltk

from nltk.stem.snowball import SnowballStemmer import regex as re

from nltk.tokenize import sent\_tokenize

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report from sklearn.model\_selection import train\_test\_split

import warnings

warnings.filterwarnings('ignore') from nltk.corpus import stopwords # download some packages

nltk.download('stopwords') nltk.download('punkt')

nltk.download('wordnet')

stop\_words = stopwords.words('english') # datasets

df\_fake = pd.read\_csv("Fake.csv") df\_true = pd.read\_csv("True.csv") df\_true.head()

df\_fake.head()

# label them seperately df\_true['status'] = 0

df\_fake['status'] = 1

# merge and remove unnecessary columns df = pd.concat([df\_true,df\_fake])

df.drop(['subject','text','date'],axis=1,inplace=True) # let's blend the smoothie

random\_indexes = np.random.randint(0,len(df),len(df)) df = df.iloc[random\_indexes].reset\_index(drop=True)

pd.set\_option('display.max\_colwidth', 500) random = np.random.randint(0,len(df),20)

df.iloc[random] # Null values

df.isnull().sum()

# longest sentence length

def longest\_sentence\_length(text): return len(text.split())

df['maximum\_length'] = df['title'].apply(lambda x : longest\_sentence\_length(x)) print('longest sentence having length -')

max\_length = max(df['maximum\_length'].values) print(max\_length)

# Text cleaning

text\_cleaning = "\b0\S\*|\b[^A-Za-z0-9]+"

def preprocess\_filter(text, stem=False):

text = re.sub(text\_cleaning, " ",str(text.lower()).strip()) tokens = []

for token in text.split():

if token not in stop\_words: if stem:

stemmer = SnowballStemmer(language='english') token = stemmer.stem(token)

tokens.append(token)

return " ".join(tokens)

# Word embedding with pre padding

def one\_hot\_encoded(text,vocab\_size=5000,max\_length = 40):

hot\_encoded = one\_hot(text,vocab\_size) return hot\_encoded

# word embedding pipeline def word\_embedding(text):

preprocessed\_text=preprocess\_filter(text) return one\_hot\_encoded(preprocessed\_text)

# Creating NN Model

embedded\_features = 40 model = Sequential()

model.add(Embedding(5000,embedded\_features,input\_length = max\_length)) model.add(LSTM(100))

model.add(Dense(1,activation='sigmoid'))

model.compile(loss = 'binary\_crossentropy',optimizer= 'adam',metrics = ['accuracy']) print(model.summary())

# One hot encoded title

one\_hot\_encoded\_title =df['title'].apply(lambda x : word\_embedding(x)).values # padding to make the size equal of the sequences

padded\_encoded\_title = pad\_sequences(one\_hot\_encoded\_title,maxlen=max\_length,padding = 'pre') # Splitting

X = padded\_encoded\_title y = df['status'].values y = np.array(y)

# shapes

print(X.shape) print(y.shape)

# shape and size

print('X shape {}'.format(X.shape))

print('y shape {}'.format(y.shape)) # Splitting into training, testing

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y, random\_state = 42)

# Shape and size of train and test dataset

print('X train shape {}'.format(X\_train.shape)) print('X test shape {}'.format(X\_test.shape))

print('y train shape {}'.format(y\_train.shape)) print('y test shape {}'.format(y\_test.shape))

# Model training # training

model.fit(X\_train,y\_train,validation\_data=(X\_test,y\_test),epochs=15,batch\_size=64) # setting threshold value

def best\_threshold\_value(thresholds:list,X\_test): accuracies = []

for thresh in thresholds:

ypred =model.predict(X\_test)

ypred = np.where(ypred> thresh,1,0)

accuracies.append(accuracy\_score(y\_test,ypred)) return pd.DataFrame({

'Threshold': thresholds, 'Accuracy' : accuracies

# Predictino value at threshold 0.4 y\_pred = model.predict(X\_test)

y\_pred = np.where(y\_pred >0.4, 1, 0) # Confusion matrix

print('Confusion matrix')

print(confusion\_matrix(y\_pred,y\_test)) print(' ')

print('Classification report')

print(classification\_report(y\_pred,y\_test))

# input generator

def prediction\_input\_processing(text): encoded = word\_embedding(text)

padded\_encoded\_title = pad\_sequences([encoded],maxlen=max\_length,padding = 'pre') output = model.predict(padded\_encoded\_title)

output = np.where(0.4>output,1,0) if output[0][0] == 1:

return 'Yes this News is fake' return 'No, It is not fake'

# predictions

prediction\_input\_processing('Americans are more concerned over Indians fake open source contribution')

news = 'Trump Just Sent Michelle Obama a Bill She will Never Be able to pay in her lifetime' prediction\_input\_processing(news)

APPENDIX

https://github.com/revanthroshan/fakenews\_prediction/upload/main