FAKE NEWS CLASSIFICATION USING MACHINE LEARNING WITH NLP

A PROJECT REPORT

Submitted by

Binishmon .B	**712218205009**
Madhankumar .R	**712218205023**
Tamilselvan .R	**712218205053**
Velavar .K	**712218205056**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PARK COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY:CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"FAKE NEWS CLASSIFICATION USING MACHINE LEARNING WITH NLP"** is the bonafide work of **"Binishmon.B, Madhankumar.R, Tamilselvan.R, Velavar.K"** who carried out the project work under my supervision.

SIGNATURE
Dr.Vijaya lakshmi

HEAD OF THE DEPARTMENT
Professor
Department of information Technology
Park college of Engineering and Technology
Kaniyur,Coimbatore-641659

SIGNATURE
Dr.Vijaya lakshmi

SUPERVISOR
Professor
Department of information Technology
Park college of Engineering and Technology
Kaniyur,Coimbatore-641659

Certified that the candidate was examined in the viva-voce examination held On_____

Internal Examiner **External Examinar**

ACKNOWLEDGEMENT

First and foremost, we praise and thank God and our parents, from the bottom of our heart for bestowing his benediction on us throughout the course of this project.

We wish to express our sincere thanks to our beloved Chairman **Dr.P.V.RAVI Ph.D., MISTE,** for providing an opportunity to work on this project.

We would like to express our deep sense of gratitude to the Chief Executive Officer **Dr. ANUSHA RAVI B.E., M.S. (USA), Ph.D,** for her valuable support and encouragement.

We extend sincere thanks to our Principal **Dr. G. MOHAN KUMAR B.E., M.E., Ph.D, MBA, PGDMRM, MIE, MISTE,** for supporting and encouraging to carry out our project.

We are highly indebted and graceful to our beloved Head of the Department and guide **Dr.S.VIJAYALAKSHMI M.E, Ph.D,** for his valuable and his constant suggestions and persistent encouragement.

We also thank all the faculty members, teaching and non-teaching staff and lab in-charge of our department who helped us to complete our project successfully

ABSTRACT

Fake news has been a problem ever since the internet boomed. The very network that allows us to know what is happening globally is the perfect breeding ground for malicious and fake news. Combating this fake news is important because the world's view is shaped by information.

People not only make important decisions based on information but also form their own opinions. If this information is false it can have devastating consequences.

Verifying each news one by one by a human being is completely unfeasible. This paper attempts to expedite the process of identification of fake news by proposing a system that can reliably classify fake news.

Newspapers are the primary source of news for people worldwide. However, off late, due to the significant growth and updates in technologies, there has been a stupendous rise in the popularity of social media. The number of people who use social media has increased remarkably. As a consequence, social networks such as social media, websites, blogs, etc. have emerged as relevant platforms to gather all kinds of news. People rely more on social networks than newspapers these days. With the availability of the internet, these networks can be accessed easily.

This can lead to easy manipulation of the existing news, thereby causing fake news. Fake news can be used as a vital tool to project people in a wrong way. It can spread hate among people which can further harm the society. Hence, it is very necessary to prevent the spread of fake news. This survey paper describes the various methods and models used for the detection of fake news. Our project aims to use Natural Language Processing to directly detect fake news, based on the text content of news articles

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

English Dictionary defines fake news as "a story that is presented as being a genuine item of news but is in fact not true and is intended to deceive people". But the definition isn't as clear cut in the 21st century.The water becomes murky as we start to question if rumors are fake news or if parodies or political humor is fake news. Can an exaggeration of a simple news article be considered fake as they may portray the subject in a different, if not in a negative view.

News media has become a channel to pass on the information of what's happening in the world to the people living. Often people perceive whatever conveyed in the news to be true. There were circumstances where even the news channels acknowledged that their news is not true as they wrote.

But some news has a significant impact not only on the people or government but also on the economy.

In the modern era, the spread of fake news has become very evident. Fake news is being used for both economic and political benefits. The need of the hour is to prevent the spread of fake news. The first thing that needs to be done to achieve this is to detect fake news. Our project aims to develop a machine learning program to identify when a news source may be producing fake news. We use a corpus of labelled real and fake articles to build a classifier that can make decisions about information based on the content from the corpus.

Our model focuses on identifying sources of fake news, based on multiple articles originating from a source. Once a source is labelled as a producer of fake news, we predict that all future articles from the same source are also a producer of fake news. The intended application of our project is to assist in applying visibility weights in social media. Social networks can make use of the weights produced by the model to obscure stories that are highly likely to be fake news.

CHAPTER 2

2.1 Existing work

" Gurbani Gulati; Bhanu Prakash Lohani; Pradeep Kumar Kushwaha" Fake news occurs when a news article is intentionally released and subsequently shown to be false. This can negatively impact on an individual's or indeed society's comprehension of topics and shape public opinion on major events, e.g. elections. With the ubiquity of social media and fake news websites, people are increasingly exposed to fake news. In this article, we explore and compare the performance of fake news detection approaches using both machine and deep learning methods. Specifically we explore Logistic Regression, Support Vector Machines (SVM), Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT) and compare the results with human crowds ability to distinguish real vs fake news. We utilise over 8.5m records from FakeNewsCorpus and 23k records from FakeNewsNet. The results shown that BERT achieved the best overall accuracy at 82.5%. Groups of individual assessors achieved an accuracy of 79%, whilst individuals varied significantly between 60-80% in their ability to distinguish real vs fake news.

” Muskan; Teena Khandelwal; Manisha Khandelwal; Purnendu Shekhar Pandey” With the changing era of digital technology and use of 5G networks it is ubiquitous to use portable devices on palm which facilitates users to access information in abundance with low cost from different social media sites. It also provides common platform for everyone to share information very fast. So veracity of such information is important otherwise it can mislead the mass and create issues. So detection of such news or contents which are false or fake is a challenging task. This paper analyses different machine learning techniques to detect whether the news contents are true or false. For this, multi-attribute relational features of the LIAR dataset are used. Sentiment features of the news statements are used along with the metadata of the speaker, which is taken as relations. Experimental result shows that the proposed method gives an accuracy of 75% compared to other techniques used on this dataset.

Gurbani Gulati; Bhanu Prakash Lohani; Pradeep Kumar Kushwaha, **“Fake news detection”** using K-NN, SVM and many more to verify the news. Further, cross-validation was used and the top three machine learning algorithms

2.1 Draw backs:

* Low Level Accuracy
* Over Fitting

2.2 Literature review:

Literature survey 1

Title: Demystifying Black-box Learning Models of Rumor Detection from Social Media Posts

Author: Faiza Tafannum; Mir Nafis Sharear Shopnil; Anika Salsabil; Navid Ahmed; Md. Golam Rabiul Alam; Md Tanzim Reza

Abstract

Social media and its users are vulnerable to the spread of rumors, therefore, protecting users from the spread of rumors is extremely important. For this reason, we propose a novel approach for rumor detection in social media that consists of multiple robust models: XGBoost Classifier, Support Vector Machine, Random Forest Classifier, Extra Tree Classifier, Decision Tree Classifier, a hybrid model, deep learning models-LSTM and BERT. For evaluation, two datasets are used. These artificial intelligence algorithms are often referred to as "Blackbox&quot Although, there have been several works on detecting fake news, the number of works regarding rumor detection is still limited and the models used in the existing works do not explain their decision-making process. We take models with higher accuracy to illustrate which feature of the data contributes the most for a post to have been predicted as a rumor or a non-rumor by the models to explain the opaque process happening inside the black-box models. Our hybrid model achieves an accuracy of 93.22% and 82.49%, while LSTM provides 99.81%, 98.41% and BERT provides 99.62%, 94.80% accuracy scores on the COVID19 Fake News and the concatenation of Twitter15 .

Literature survey 2

Title: Identification of Fake News Using Deep Learning Architecture

Author : R. Mahesh; Bonam Poornika; Nimma Sharaschandrika;

There is extensive increase of false news may be produced by human beings, computers or any automatic machines. This increase has negative influence on both society and individuals in both political and social perspectives. In this contemporary world of social media and connected networks, evaluating reliability promptly is really challenging by the rapid spread of news. As a result, automatically detecting false news techniques has a huge demand in current scenario. In order to overcome prior issuance, this research work has come up with an integrated hybrid neural network architecture, which consists of both the features of CNN along with LSTM and also some dimensionality reduction techniques such as principal component analysis and chi-square. The dimensionality reduction method is used to decrease the dimensionality of feature vector. Also, the dimensionality reduction is used before passing them to classifier. Also, a dataset has been acquired for training and testing the proposed model from a website named Fake news challenge. This has two kinds of outputs: real and fake. These features provide additional contextual and are given to PCA and Chi-Square for faux information detection. The motive related to this project is discovering news article position towards headline.

Literature survey 3

Title: Beyond fear go viral: A machine learning study on infodemic detection during covid-19 pandemic

Author : Tipajin Thaipisutikul; Timothy K. Shih; Avirmed Enkhbat; Wisnu Aditya; Huang-Chia Shih; Pattanasak Mongkolwat

Abstract

With the restrictions in our daily life activities under the current situation of the covid-19 pandemic worldwide, billions of people rely on social media platforms to share and obtaining covid-19 related news information. This made social media platforms easily be used as a source of myths and disinformation, which can cause severe public risks. It is thus of vital importance to constraint the spread of misinformation to the public. Although many works have shown promising results on the misinformation detection problem, only a few studies focus on the infodemic detection during the covid-19 pandemic, especially in the low resource language like Thai. Therefore, in this paper, we conduct extensive experiments on the real-world social network datasets to detect misinformation about covid-19 targeting both English and Thai languages. In particular, we perform an exploratory data analysis to get the statistic and characteristics of real and fake content. Also, we evaluate a series of three feature extraction, seven traditional machine learning, and eleven deep learning methods in detecting the fabricated content on social media platforms. The experimental results demonstrate that the transformer-based model significantly outperforms other deep learning and traditional machine learning methods in all metrics, including accuracy and F-measure.

Literature survey4

Title: BERT Model for Classification of Fake News using the Cloud Processing Capacity

Author : Raj Mohan; C. Vipin Raj; P. Aswathi; Rao R. Bhavani

Abstract

This paper aims at conducting a predictive analysis on news articles in order to find if they are fake or real. After conducting an extensive research on the topic, various Machine Learning and Deep Learning models for the purpose of evaluating news articles were discovered. A new transfer learning model, Bi-directional Encoder Representation for Transformers (BERT), is tested using the Google Cloud GPU capacity for the purpose of detection. The first step in this direction will be to pre-process the data to clean out the garbage and missing values. After this, all the news articles collected will be tokenized, according to the BERT tokenizer. The tokenized corpus will be converted into tensors for the model to be trained. The data will be trained in batches with each batch having 32 articles. The final layer for training will consist of a five layered neural network. The model with the least validation loss will be tested for accuracy. The predictions for news articles will be made on this model. The paper will also explore the best cloud platform to host such a model and performance of the hosted model as well.

Literature survey 5

Title: A Performance Comparison of Fake News Detection Approaches

Author : Gurbani Gulati; Bhanu Prakash Lohani; Pradeep Kumar Kushwaha

Abstract

Fake news occurs when a news article is intentionally released and subsequently shown to be false. This can negatively impact on an individual's or indeed society's comprehension of topics and shape public opinion on major events, e.g. elections. With the ubiquity of social media and fake news websites, people are increasingly exposed to fake news. In this article, we explore and compare the performance of fake news detection approaches using both machine and deep learning methods. Specifically we explore Logistic Regression, Support Vector Machines (SVM), Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT) and compare the results with human crowds ability to distinguish real vs fake news. We utilise over 8.5m records from FakeNewsCorpus and 23k records from FakeNewsNet. The results shown that BERT achieved the best overall accuracy at 82.5%. Groups of individual assessors achieved an accuracy of 79%, whilst individuals varied significantly between 60-80% in their ability to distinguish real vs fake news.

Literature survey6

Title: Classification of News Articles with Relational Multi Attributes using Machine Learning

Author : Muskan; Teena Khandelwal; Manisha Khandelwal; Purnendu Shekhar Pandey

Abstract

With the changing era of digital technology and use of 5G networks it is ubiquitous to use portable devices on palm which facilitates users to access information in abundance with low cost from different social media sites. It also provides common platform for everyone to share information very fast. So veracity of such information is important otherwise it can mislead the mass and create issues. So detection of such news or contents which are false or fake is a challenging task. This paper analyses different machine learning techniques to detect whether the news contents are true or false. For this, multi-attribute relational features of the LIAR dataset are used. Sentiment features of the news statements are used along with the metadata of the speaker, which is taken as relations. Experimental result shows that the proposed method gives an accuracy of 75% compared to other techniques used on this dataset.

CHAPTER 3

3.PROPOSED SYSTEM

RNN, and This work is mainly concentrated on using pure NLP perspective to identify the presence of fake news by utilizing the linguistic features LSTM(long short-term memory) which helps in containing sequence information. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

Term frequency-inverse document frequency (TF-IDF) for Word Extraction and Feature Selection

NLP (Natural Language Processing) With Text Analysis

The proposed system when subjected to a scenario of a set of news articles , the new articles are categorized as true or fake by the existing data available . This prediction is done by using the relationship between the words used in the article with one another. The proposed system contains a Word2Vec model for finding the relationship between the words and with the obtained information of the existing relations , the new articles are categorized into fake and real news.

Input is collected from various sources such as newspapers , social media and stored in datasets. System will take input from datasets. The datasets undergo preprocessing and the unnecessary information is removed from it and the data types of the columns are changed if required. Jupyter notebook and python libraries are used in the above step. Count vectorizer technique is used in the initial step. For fake news detection , we have to train the system using dataset. Before entering to the detection of fake news , entire dataset is divide into two datasets . 80% is used for training and 20% is used for testing. During

training , K-Means algorithm is used to train the model using the train dataset. In testing , the test dataset is given as input and the output is predicted.After the testing time , The predicted output and the actual output are compared using confusion matrix obtained .The confusion matrix gives the information regarding the number of correct and wrong predictions in the case of real and fake news.The accuracy is calculated by the equation No Of Correct Predictions/Total Test Dataset Input Size

Algorithms for proposed system:

Step 1: Start

Step 2: Input is collected from various sources and prepare a dataset.

Step 3: Preprocessing of data is done and dataset is divided into 2 parts training and testing data.

Step 4: Count vectorization technique is used to convert the train data into numericals.

Step 5: K MEANS clustering algorithm is used to build the predictive model using the train data . Step 6: Confusion matrix is obtained . Step 7: Accuracy is calculated.

3.1 Concept:

This is an important phase for any text analysis application. There will be much un-useful content in the news which can be an obstacle when feeding to a machine learning model.

3.2 Technology:

- **Machine learning:**

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." This is Alan Turing's definition of machine learning. Deep learning is a class of machine learning algorithms that utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach. The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited. Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields

including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

- **NLP(Natural language processing):**

  NATURAL LANGUAGE PROCESSING is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data. Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyse large amounts of natural language data.

3.3 Deep learning-LSTM:

We aren't gonna use a normal neural network like ANN to classify but LSTM(long short-term memory) which helps in containing sequence information. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems

3.4 K-Nearest Neighbor :

In machine learning, K-Nearest Neighbor is one of the popular Supervised technique used for different regression and classification problems .

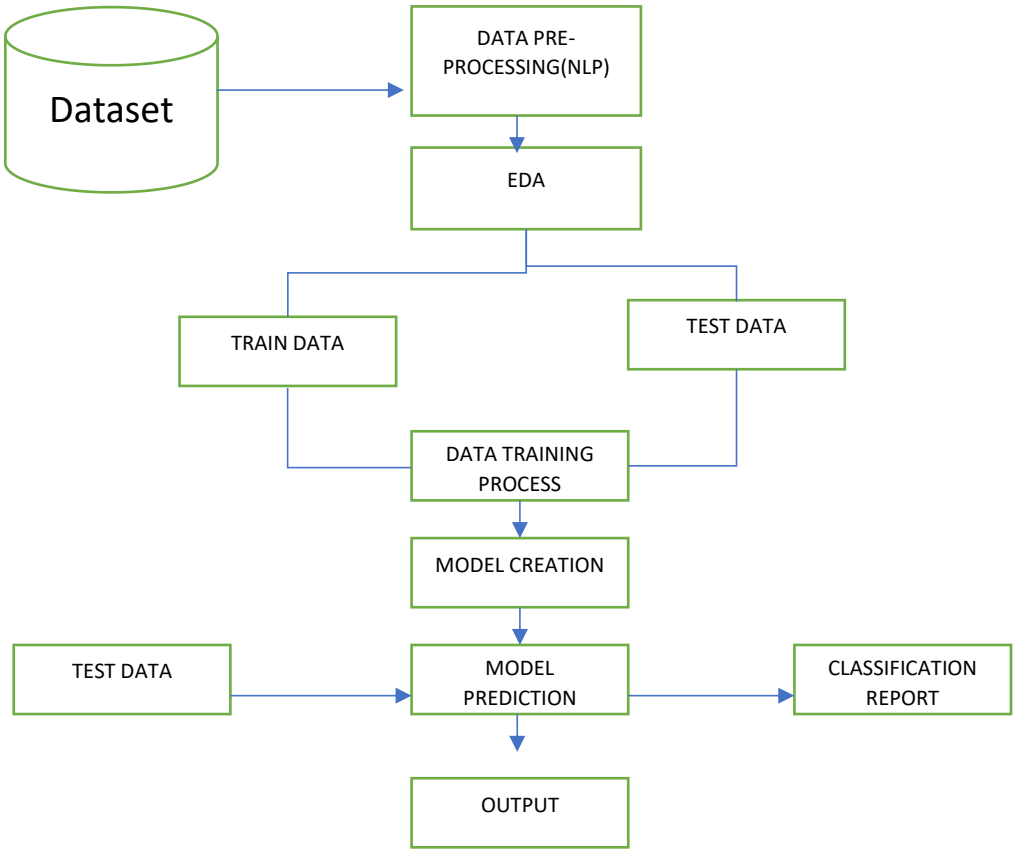
Supervised Learning is a learning that is executed using label data points. KNN classification can be effectively used as an anomaly detection technique (i.e. fraud); KNN regression can be applied in many types of regression problem effectively, including actuarial models, environmental model and real estate model. Now we can also use this algorithm for Fake News Detection on Social media K-Nearest Neighbor is a simple technique for building a classification model which assign the class label to problem instances. This is not only one algorithm for training such classifier but a family of algorithm exist. KNN is used in recommendation systems, semantic searching and detection and anomaly detection. It's mostly used for classification problem in the industry as it's easy to interpret, consume less time and can easily handle the noise. In machine learning problems, the first thing is finding a way to represent data points feature vector. A feature vector is a mathematical representation of data. If a given data is having N unique feature than the length of the feature vector is N. The aim of this algorithm is to classify a new object based on their training data and attributes.

To classify new object KNN performs some steps.

• Evaluate the distance among the item to be classified and each point in the training dataset.

• Pick the closest data points with the K lowest distance.

3.5 Flow diagram:



3.6 Problem Statement:

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

The sensationalism of not-so-accurate eye-catching and intriguing headlines aimed at retaining the attention of audiences to sell information has persisted all throughout the history of all kinds of information broadcast. On social networking websites, the reach and effects of information spread are however significantly amplified and occur at such a fast pace, that distorted, inaccurate, or false information acquires a tremendous potential to cause real impacts, within minutes, for millions of users.

3.7 Advantages:

- RNN can process inputs of any length.
- An RNN model is modeled to remember each information throughout the time which is very helpful in any time series predictor.
- Even if the input size is larger, the model size does not increase

.

CHAPTER 4

4.SYSTEM IMPLEMENTATION

4.1 Modules:

1.Data Collection

2.Pre-processing and Cleaning

3.Training and Testing Data

4.Model Create

5.Model Prediction

4.1.1 Data Colletion:

This metadata has 2 CSV files where one dataset contains fake news and the other contains true/real news and has nearly 23481 fake news and 21417 true news

Description of columns in the file:

1) title- contains news headlines

2) text- contains news content/article

3) subject- the type of news

4) date- the date the news was published

We can get online news from different sources like social media websites, search engine, homepage of news agency websites or the fact-

checking websites. On the Internet, there are a few publicly available datasets for Fake news classification like Buzzfeed News, LIAR

BS Detector etc. These datasets have been widely used in different research papers for determining the veracity of news. In the

following sections, I have discussed in brief about the sources of the dataset used in this work.

Online news can be collected from different sources, such as news agency homepages, search engines, and social media websites.

However, manually determining the veracity of news is a challenging task, usually requiring annotators with domain expertise who

performs careful analysis of claims and additional evidence, context, and reports from authoritative sources.

Generally, news data with annotations can be gathered in the following ways: Expert journalists, Fact-checking websites, Industry detectors, and Crowd sourced workers. However, there are no agreed upon benchmark datasets for the fake news detection problem. Data gathered must be pre-processed- that is, cleaned, transformed and integrated before it can undergo training process

Fake news samples :

contains a dataset of fake news articles that was gathered by using a tool called the BS detector which essentially has a blacklist of websites that are sources of fake news. The articles were all published in the 30 days between October, 26 2016 to November 25, 2016. While any span of dates would be characterized by the current events of that time, this range of dates is particularly interesting because it spans the time directly before, during, and directly after the 2016 election. The dataset has articles and metadata from 244 different websites, which is helpful in the sense that the variety of sources will help the model to not learn a source bias. However, at a first glance of the dataset, you can easily tell that there are still certain obvious reasons that a model could learn specifics

of what is included in the “body” text in this dataset. For example, there are instances of the author and source in the body text, Also, there are some patterns like including the date that, if not also repeated in the real news dataset, could be learned by the model.

Real news samples :

As suggested by an acceptable approach would be to use the APIs from reliable sources like New York Times and The Guardian. The NYT API provides similar information to that of the kaggle dataset, including both text and images that are found in the document. The Kaggle Dataset also provides the source of each article, which is trivial for the APIs of specific newspaper sources. We 17 pulled articles from both of these sources in the same range of dates that the fake news was restricted to (October 26 , 2016 to November 25, 2016). This is important because of the specificity of the current events at that time - information that would not likely be present in news outside of this timeframe. There were just over 9,000 Guardian articles and just over 2,000 New York Times articles. Unlike the Kaggle dataset, which had 244 different websites as sources, our real news dataset only has two different source: The New York Times and The Guardian.

Due to this difference, we found that extra effort was required to ensure that we removed any source-specific patterns so that the model would not simply learn to identify how an article from the New York Times is written or how an article from The Guardian is written. Instead, we wanted our model to learn more meaningful language patterns that are similar to real news reporting, regardless of the source

4.1.2 Pre-processing and Cleaning:

Text Processing:

This is an important phase for any text analysis application. There will be much un-useful content in the news which can be an obstacle when feeding to a machine learning model. Unless we remove them the machine learning model doesn't work efficiently.

News-Punctuation Cleaning

Punctuation is the tool that allows us to organize our thoughts and make it easier to review and share our ideas. The standard English punctuation is as follows: period, comma, apostrophe, quotation, question, exclamation, brackets, braces, parenthesis, dash, hyphen, ellipsis, colon, semicolon.

News-Stop words

Punctuation is the tool that allows us to organize our thoughts and make it easier to review and share our ideas. The standard English punctuation is as follows: period, comma, apostrophe, quotation, question, exclamation, brackets, braces, parenthesis, dash, hyphen, ellipsis, colon, semicolon.

News-Stop words

A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words to take up space in our database, or taking up the valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words

4.1.3 Training data and Test data :

a) For choosing a model we split our dataset into train and test

b) Here data's are split into 3:1 ratio that means

c) Training data having 70 percent and testing data having 30 percent

d) In this split process preforming based on train_test_split model

e) After splitting we get xtrainxtest and ytrainytest

4.1.4 Model Creation :

a) Contextualise machine learning in your organisation.

b) Explore the data and choose the type of algorithm.

c) Prepare and clean the dataset.

d) Split the prepared dataset and perform cross validation.

e) Perform machine learning optimisation.

f)Deploy the model.

4.1.5 Model Prediction:

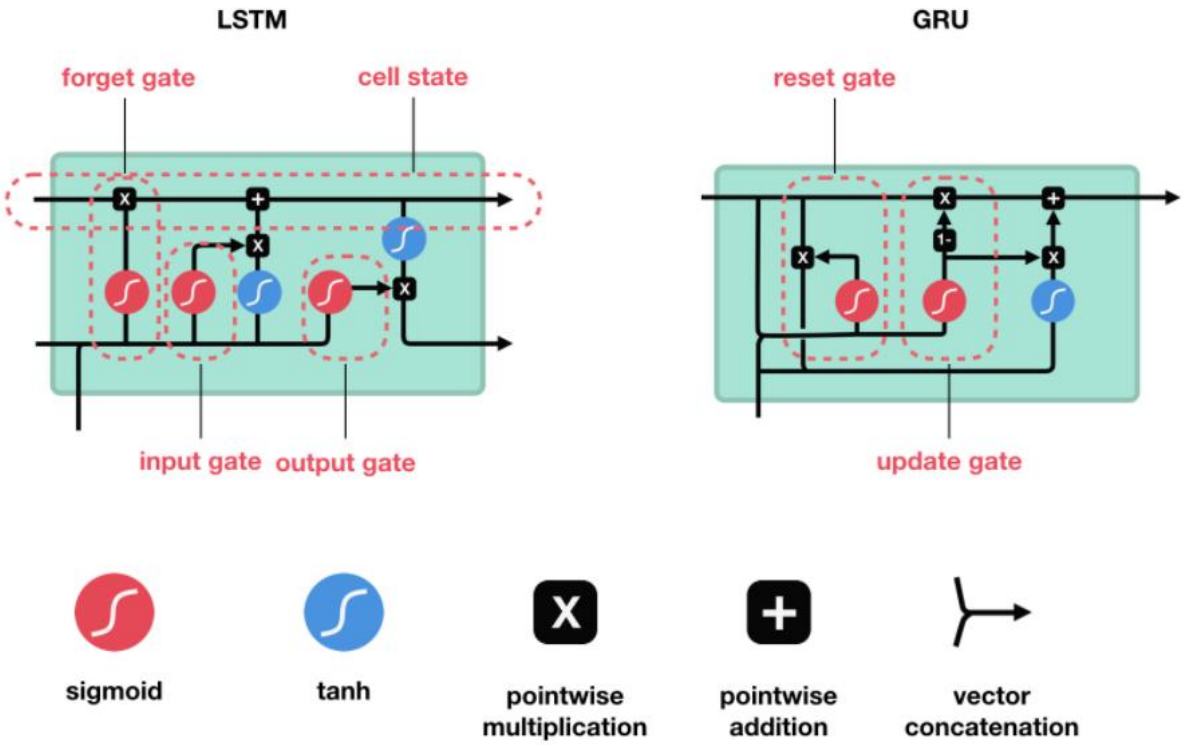
Predictive modeling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data. It works by analyzing current and historical data and projecting what it learns on a model generated to forecast likely outcomes.

4.2 Padding embedded documents:

All the neural networks require to have inputs that have the same shape and size. However, when we pre-process and use the texts as inputs for our LSTM model, not all the sentences have the same length. In other words, naturally, some of the sentences are longer or shorter. We need to have the inputs of the same size, this is where the padding is necessary. Here we take the common length as 5000 and perform padding using pad_sequence() function. Also, we are going to ‘pre’ pad so that zeros are added before the sentences to make the sentence of equal length

4.3 LSTM Model:

Long Short-Term Memory networks (LSTM) are a special type of RNN competent in learning long-term dependencies . LSTM is a very effective solution for addressing the vanishing gradient problem. In LSTM-RNN the hidden layer of basic RNN is replaced by an LSTM cell



Text Processing

1.NLP With Stop words Removing

2.Word Cloud of Fake and True News

3.Time series analysis- Fake/True news

4.Stemming words Removing

5.Algorithm Implementation

4.4 Bi-directional Long Short-Term Memory - Recurrent Neural Network

LSTMs help to preserve the error that can be back-propagated through time and in lower layers of a deep network Bi-directional processing is an evident approach for a large text sequence prediction and text classification. a Bi-Directional LSTM network steps through the input sequence in both directions at the same time.

The proposed fake news detection model based on Bi-directional LSTM-recurrent neural network. The news articles are first pre-processed. A binary label is set to each news article as 1 for fake news and 0 for real news. The input news articles are converted to UTF-8 format and punctuation and stop words are removed.

The news articles' title and content text are turned into space-separated padded sequences of words. These sequences are further split into lists of tokens.

Global Vectors for Word Representation (GloVe) embeddings is provided by Stanford NLP team It is an unsupervised learning algorithm for obtaining vector representations for words. Pre-trained GloVe word embeddings are used to deal with the high dimensional news articles. The embedding layer will load the weights from GloVe instead of loading random weights. GloVe applies globally aggregated co-occurrence statistics across all words in the news article corpus. The resulting representations formalize significant linear substructures of the word vector space. The transformed vector represented data is partitioned into train, validation and test data.

The training is carried out on the news article corpus. Validation data set is used for fine-tuning the model. Further, the test data is used to know the predicted label of news article based on trained model. The model selection among CNN, variation of RNN as Vanilla RNN, LSTM-RNN, and Bi-directional LSTM-RNN is carried out to detect fake news. If CNN is selected

the model, each embedding layer corresponding to training data is fed into CNN. Different filter sizes are used for evaluation of CNN performance. For example, if the filter size is 3, the filter will stride through the document computing the calculation above with all the trigrams. Global Max Pooling layer is used to extract the maximum value from each filter. The resultant is passed through several dense hidden layers with dropout. Finally softmax layers are used to make a binary decision of whether or not the article is credible. Similarly in Bi-Directional LSTM network each embedding layer corresponding to training data is inspected in both orders at the same time.

Model is trained iteratively to minimize loss function and to improve accuracy. The cross-entropy loss is considered to detect fake news article in proposed model. Various adaptive learning optimization algorithms namely AdaGrad, RMSProp, and Adam are considered to improve the model performance.

4.5 CONFUSION MATRIX

A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

A confusion matrix is a summary of prediction results on a classification problem.

The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

The confusion matrix shows the ways in which your classification model is confused when it makes predictions.

It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being

	Class 1 Predicted	*Class 2 Predicted*
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

made.

Here,

• Class 1 : Positive

• Class 2 : Negative

Definition of the Terms:

• Positive (P) : Observation is positive (for example: is an apple).

• Negative (N) : Observation is not positive (for example: is not an apple).

• True Positive (TP) : Observation is positive, and is predicted to be positive.
• False Negative (FN) : Observation is positive, but is predicted negative.
• True Negative (TN) : Observation is negative, and is predicted to be negative.
• False Positive (FP) : Observation is negative, but is predicted positive.

Classification Rate/Accuracy:

Classification Rate or Accuracy is given by the relation:

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

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

Recall:

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (small number of FN).

Recall is given by the relation:

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

Precision:

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (small number of FP).

Precision is given by the relation:

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

High recall, low precision: This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

Low recall, high precision: This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)

F-measure:

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

The F-Measure will always be nearer to the smaller value of Precision or Recall.

$$F - measure = \frac{2*Recall*Precision}{Recall + Precision}$$

in which we have infinite data elements of class B and a single element of class A and the model is predicting class A against all the instances in the test data.

Here,

Precision : 0.0

Recall : 1.0

CHAPTER 5

5.SYSTEM REQUIREMENT

5.1 Software :

a) Python 3.9.6

b) Excel(CSV)

5.2 Hard ware :

a) Ram:4GB or 8GB

b) Hard Disk 64GB

5.3 Tools:

5.3.1 Numpy:

NumPy is a Python package. It stands for 'Numerical Python'. It is a library

Using NumPy, a developer can perform the following operations −

Mathematical and logical operations on arrays.

Fourier transforms and routines for shape manipulation.

Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

5.3.2 Matplotlib:

It is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

5.3.3 Pandas:

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics,

Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

5.3.4 Sklearn:

Scikit-learn is a machine learning library for Python. It features several regression, classification and clustering algorithms including SVMs, gradient boosting, k-means, random forests and DBSCAN. It is designed to work with Python Numpy and SciPy . The scikit-learn project kicked off as a Google Summer of Code (also known as GSoC) project by David Cournapeau as scikits.learn. It gets its name from "Scikit", a separate third-party extension to SciPy. Python Scikit-learn Scikit is written in Python (most of it) and some of its core algorithms are written in Cython for even better performance. Scikit-learn is used to build models and it is not recommended to use it for reading, manipulating and summarizing data as there are better frameworks available for the purpose. It is open source and released under BSD license.

5.3.5 Keras:

Keras is a high-level neural networks API, capable of running on top of Tensorflow , Theano, and CNTK . It enables fast experimentation through a high level, user-friendly, modular and extensible API. Keras can also be run on both CPU and GPU.Keras was developed and is maintained by Francois Chollet and is part of the Tensorflow core, which makes it Tensorflows preferred high-level API.

5.3.6 Tensorflow:

TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks.

CHAPTER 6

6.1CONCLUSION

: We have done mainstream work on processing the data and building the model. We could have indulged in changing the ngrams while vectorizing the text data. We took 2 words and vectorized them.better results with the help of LSTM network Find Out The Fake New Classification

6.3 Future Work :

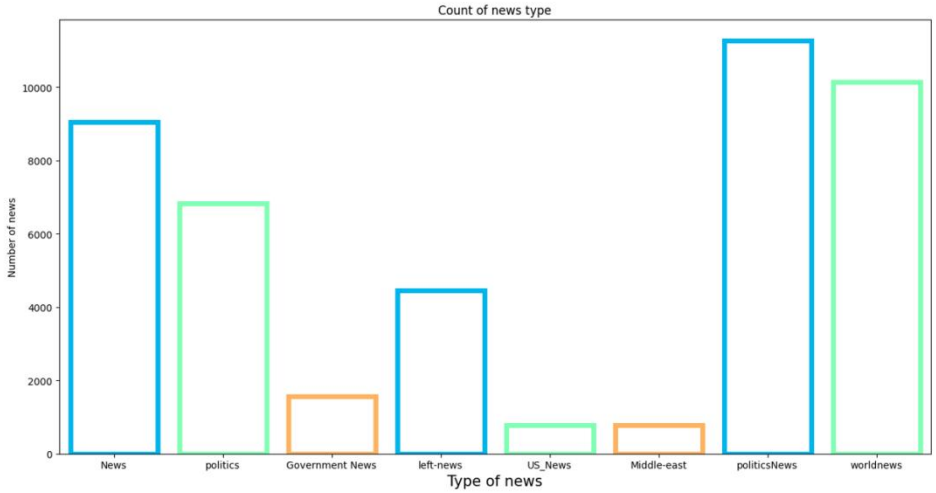
Through the work done in this project, we have shown that machine learning certainly does have the capacity to pick up on sometimes subtle language patterns that may be difficult for humans to pick up on. The next steps involved in this project come in three different aspects. The first of aspect that could be improved in this project is augmenting and increasing the size of the dataset. We feel that more data would be beneficial in ridding the model of any bias based on specific patterns in the source. There is also question as to weather 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 or not the dataset is representative of how difficult the task of separating fake from real news is.

1. We want to use web scraping and get the data from various social media and websites by ourself and use them in our system.

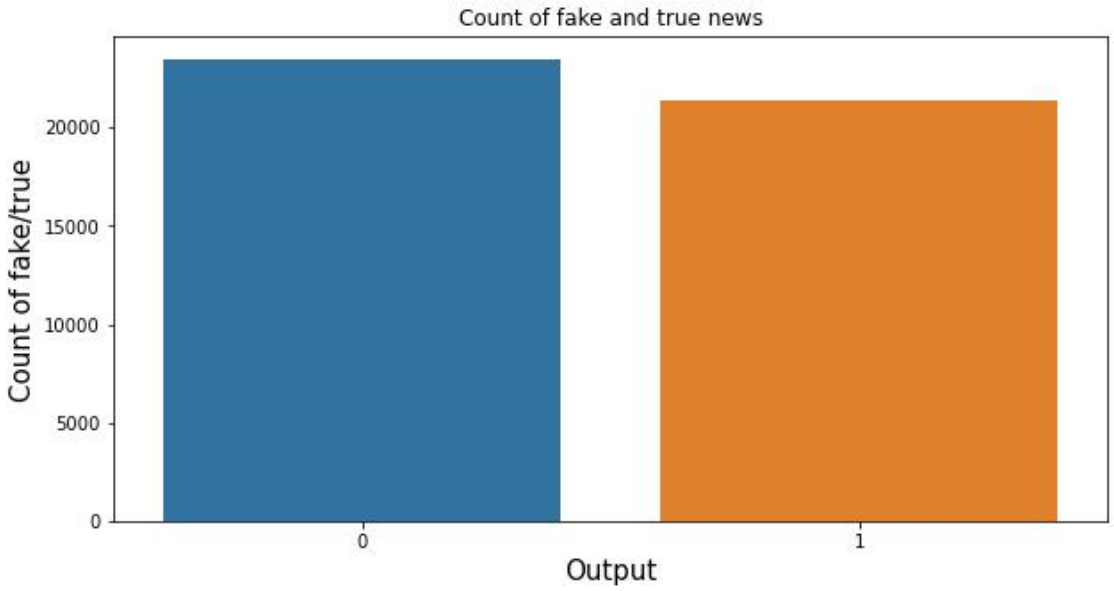
2. We also want to improve the accuracy by query optimisation

SCREENSHOTS:

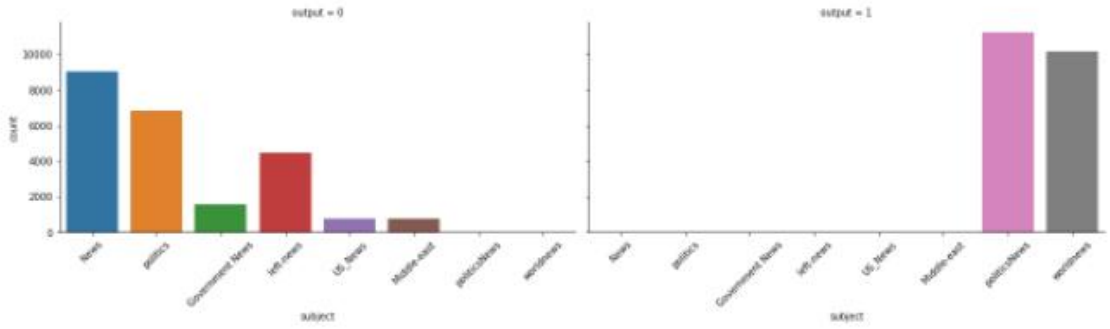
Let’s start by looking at the count of news types in our dataset



Let’s check the count of fake and true news and confirm whether our data is balanced or not



Let’s look at the count based on the fake/true outcome



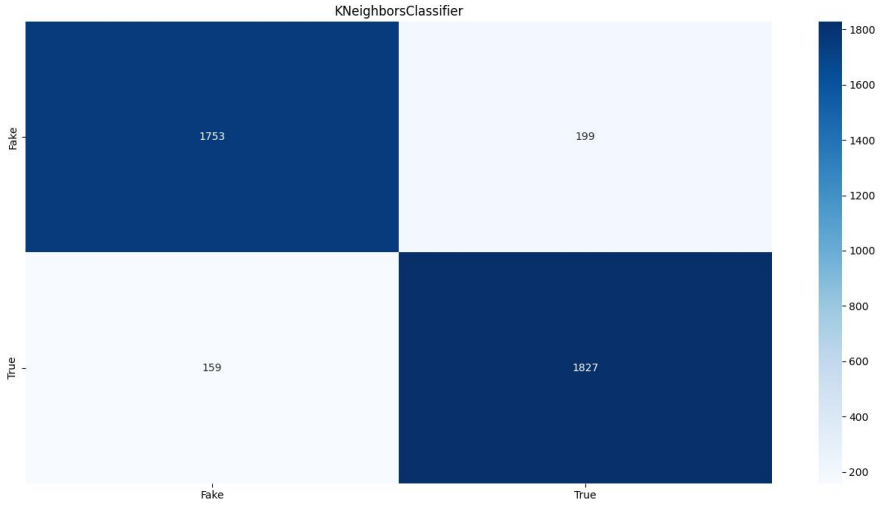
.

KNN Performance :

```
KNN Accuracy 0.9090909090909091
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.90      0.91      1952
           1       0.90      0.92      0.91      1986

    accuracy                           0.91      3938
   macro avg       0.91      0.91      0.91      3938
weighted avg       0.91      0.91      0.91      3938
```

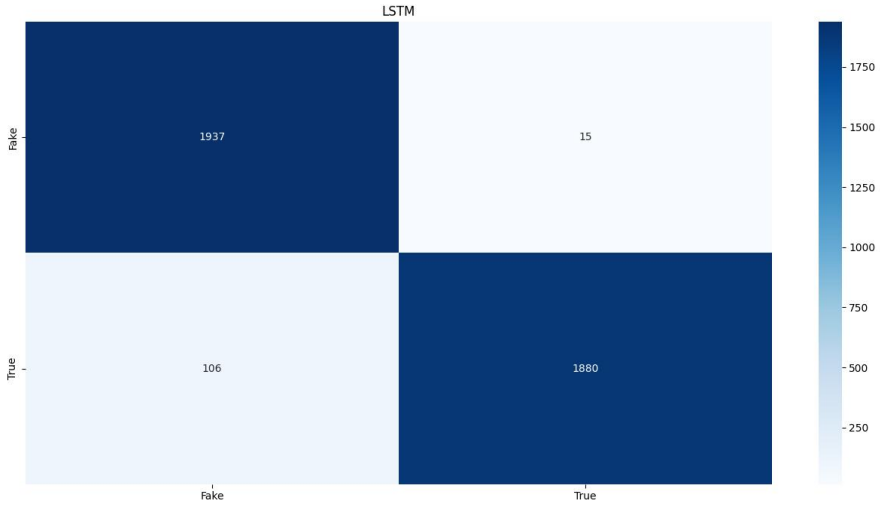


LSTM Performance :

```
LSTM Accuracy 0.9692737430167597
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      1952
           1       0.99      0.95      0.97      1986

    accuracy                           0.97      3938
   macro avg       0.97      0.97      0.97      3938
weighted avg       0.97      0.97      0.97      3938
```



Final Output:


Fake News
Detector
Predict
Looking Fake News


Fake News
Detector
Predict
Looking Real News

REFERENCE

A) Ammara Habib, Muhammad Zubair Asghar, Adil Khan, Anam Habib, Aurangzeb Khan, “False information detection in online content and its role in decision making: a systematic literature review”, Springer-Verlag GmbH Austria, part of Springer Nature 2019.

B) Thota, Aswini; Tilak, Priyanka; Ahluwalia, Simrat; and Lohia, Nibrat (2018) "Fake News Detection: A Deep Learning Approach," SMU Data Science Review: Vol. 1 : No. 3 , Article 10.

C) Chaowei Zhang, Ashish Gupta, Christian Kauten, Amit V Deokar, Xiao Qin, “Detecting fake news for reducing misinformation risks using analytics approaches ”, European Journal of Operational Research Elsevier 279 (2019).

D) Dinesh Kumar Vishwakarma, Deepika Varshney, Ashima Yadav, “Detection and veracity analysis of fake news via scrapping and authenticating the web search”, Cognitive Systems Research 58.

E) [Hossein Jahankhani, ThulasirajhJayaraveendran, and William Kapuku-Bwabw, “Improved Awareness on Fake Websites and Detecting Techniques”, ICGS3/e-Democracy 2011, LNICST 99.

APPENDIX

Source code

```
#Basic libraries
import pandas as pd
import numpy as np
#Visualization libraries
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
#NLTK libraries
import nltk
nltk.download('stopwords')
import re
import string
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from collections import Counter
#reading the fake and true datasets
fake_news = pd.read_csv('Fake.csv')
true_news = pd.read_csv('True.csv')
print('Total number of true news:',len(true_news))
print('Total number of fake news:',len(fake_news))
print('Ratio of fake news to total number news:',round(len(true_news) / (len(true_news)+len(fake_news)), 2))
#Target variable for fake news
```

```
fake_news['output']=0
#Target variable for true news
true_news['output']=1
#Concatenating and dropping for fake news
fake_news['news']=fake_news['title']+fake_news['text']
fake_news=fake_news.drop(['title', 'text'], axis=1)
#Concatenating and dropping for true news
true_news['news']=true_news['title']+true_news['text']
true_news=true_news.drop(['title', 'text'], axis=1)
#Rearranging the columns
fake_news = fake_news[['subject', 'date', 'news','output']]
true_news = true_news[['subject', 'date', 'news','output']]
fake_news['date'].value_counts()
#Removing links and the headline from the date column
fake_news=fake_news[~fake_news.date.str.contains("http")]
fake_news=fake_news[~fake_news.date.str.contains("HOST")]
'"You can also execute the below code to get the result
which allows only string which has the months and rest are filtered"'
#fake_news=fake_news[fake_news.date.str.contains("Jan|Feb|Mar|Apr|May|Jun|
Jul|Aug|Sep|Oct|Nov|Dec")]
#Converting the date to datetime format
fake_news['date'] = pd.to_datetime(fake_news['date'])
true_news['date'] = pd.to_datetime(true_news['date'])
frames = [fake_news, true_news]
news_dataset = pd.concat(frames)
news_dataset
#Creating a copy
clean_news=news_dataset.copy()
```

```
def review_cleaning(text):
    '''Make text lowercase, remove text in square brackets,remove links,remove punctuation
    and remove words containing numbers.'''
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
#Data Cleaning
clean_news['news']=clean_news['news'].apply(lambda x:review_cleaning(x))
clean_news.head()
stop = stopwords.words('english')
#stopwords Remove
clean_news['news'] = clean_news['news'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
clean_news.head()
#Plotting the frequency plot
ax = sns.countplot(x="subject", data=clean_news,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("rainbow", 3))
#Setting labels and font size
ax.set(xlabel='Type of news', ylabel='Number of news',title='Count of news type')
```

```
ax.xaxis.get_label().set_fontsize(15)
plt.show()
ax.yaxis.get_label().set_fontsize(15)
plt.show()
g = sns.catplot(x="subject", col="output",
                data=clean_news, kind="count",
                height=4, aspect=2)
#Rotating the xlabels
g.set_xticklabels(rotation=45)
plt.show()
ax=sns.countplot(x="output", data=clean_news)
#Setting labels and font size
ax.set(xlabel='Output', ylabel='Count of fake/true',title='Count of fake and true news')
ax.xaxis.get_label().set_fontsize(15)
ax.yaxis.get_label().set_fontsize(15)
plt.show()
```

MODEL:

```
import pandas as pd
import numpy as np
#Visualization libraries
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
from textblob import TextBlob
##from plotly import tools
##import plotly.graph_objs as go
##from plotly.offline import iplot
```

```
plt.rcParams['figure.figsize'] = [10, 5]
##import cufflinks as cf
##cf.go_offline()
##cf.set_config_file(offline=False, world_readable=True)
#NLTK libraries
import nltk
nltk.download('stopwords')
import re
import string
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
# Machine Learning libraries
import sklearn
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
#Metrics libraries
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.metrics import roc_auc_score
##from sklearn.metrics import roc_curve
from sklearn.metrics import confusion_matrix,plot_confusion_matrix
from sklearn.metrics import accuracy_score
#Miscellanous libraries
```

```
from collections import Counter
#Ignore warnings
import warnings
warnings.filterwarnings('ignore')
#reading the fake and true datasets
fake_news = pd.read_csv('Fake0.csv')
true_news = pd.read_csv('True0.csv')
# print shape of fake dataset with rows and columns and information
print ("The shape of the  data is (row, column):"+ str(fake_news.shape))
print (fake_news.info())
print("\n ---------------------------------------- \n")
# print shape of true dataset with rows and columns and information
print ("The shape of the  data is (row, column):"+ str(true_news.shape))
print (true_news.info())
#Target variable for fake news
fake_news['output']=0
#Target variable for true news
true_news['output']=1
#Concatenating and dropping for fake news
fake_news['news']=fake_news['title']+fake_news['text']
fake_news=fake_news.drop(['title', 'text'], axis=1)
#Concatenating and dropping for true news
true_news['news']=true_news['title']+true_news['text']
true_news=true_news.drop(['title', 'text'], axis=1)
#Rearranging the columns
fake_news = fake_news[['subject', 'date', 'news','output']]
true_news = true_news[['subject', 'date', 'news','output']]
```

```
fake_news['date'].value_counts()
#Removing links and the headline from the date column
fake_news=fake_news[~fake_news.date.str.contains("http")]
fake_news=fake_news[~fake_news.date.str.contains("HOST")]
#Converting the date to datetime format
fake_news['date'] = pd.to_datetime(fake_news['date'])
true_news['date'] = pd.to_datetime(true_news['date'])
frames = [fake_news, true_news]
news_dataset = pd.concat(frames)
news_dataset
#Creating a copy
clean_news=news_dataset.copy()
def review_cleaning(text):
    '''Make text lowercase, remove text in square brackets,remove links,remove punctuation
    and remove words containing numbers.'''
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
clean_news['news']=clean_news['news'].apply(lambda x:review_cleaning(x))
clean_news.head()
stop = stopwords.words('english')
```

```
clean_news['news'] = clean_news['news'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))

clean_news.head()

stop = stopwords.words('english')

clean_news['news'] = clean_news['news'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))

clean_news.head()

#Extracting 'reviews' for processing

news_features=clean_news.copy()

news_features=news_features[['news']].reset_index(drop=True)

news_features.head()

stop_words = set(stopwords.words("english"))

#Performing stemming on the review dataframe

ps = PorterStemmer()

#splitting and adding the stemmed words except stopwords

corpus = []

for i in range(0, len(news_features)):

    news = re.sub('[^a-zA-Z]', ' ', news_features['news'][i])

    news= news.lower()

    news = news.split()

    news = [ps.stem(word) for word in news if not word in stop_words]

    news = ' '.join(news)

    corpus.append(news)

tfidf_vectorizer = TfidfVectorizer(max_features=5000,ngram_range=(2,2))

# TF-IDF feature matrix

X= tfidf_vectorizer.fit_transform(news_features['news'])

X.shape

#Getting the target variable
```

```
y=clean_news['output']
## Divide the dataset into Train and Test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
KNN = KNeighborsClassifier()
KNN.fit(X_train, y_train)
y_pred = KNN.predict(X_test)
KNN = accuracy_score(y_test,y_pred)
print("KNN Accuracy",KNN)
matrix = confusion_matrix(y_test,y_pred)
print("Classification Report:\n",classification_report(y_test, y_pred))
categories = ['Fake','True']
df_matrix = pd.DataFrame(matrix, index = categories, columns = categories)
sns.heatmap(df_matrix, annot=True, cmap='Blues', fmt='d')
plt.title('KNeighborsClassifier')
plt.show()
#Ignore warnings
import warnings
warnings.filterwarnings('ignore')
#Deep learning libraries
from tensorflow.keras.layers import Embedding
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
corpus[1]
```

```
#Setting up vocabulary size
voc_size=10000
#One hot encoding
onehot_repr=[one_hot(words,voc_size)for words in corpus]
#Setting sentence length
sent_length=5000
#Padding the sentences
embedded_docs=pad_sequences(onehot_repr,padding='pre',maxlen=sent_length)
print(embedded_docs)
embedded_docs[1]
#Creating the lstm model
embedding_vector_features=40
model=Sequential()
model.add(Embedding(voc_size,embedding_vector_features,input_length=sent
_length))
model.add(Dropout(0.3))
model.add(LSTM(100)) #Adding 100 lstm neurons in the layer
model.add(Dropout(0.3))
model.add(Dense(1,activation='sigmoid'))
#Compiling the model
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy']
)
print(model.summary())
len(embedded_docs),y.shape
# Converting the X and y as array
X_final=np.array(embedded_docs)
y_final=np.array(y)
#Check shape of X and y final
```

```
X_final.shape,y_final.shape
# Train test split of the X and y final
X_train, X_test, y_train, y_test = train_test_split(X_final, y_final, test_size=0.33, random_state=42)
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=100,batch_size=64)
# Predicting from test data
y_pred=model.predict(X_test)
print(classification_report(y_pred,y_test))
LSTM=accuracy_score(y_pred,y_test)
print("LSTM",LSTM)
matrix = confusion_matrix(y_pred,y_test)
print("Classification Report:\n",classification_report(y_test, y_pred))
categories = ['Fake','True']
df_matrix = pd.DataFrame(matrix, index = categories, columns = categories)
sns.heatmap(df_matrix, annot=True, cmap='Blues', fmt='d')
plt.title('KNeighborsClassifier')
plt.show()
scores = [KNN,LSTM]
algorithms = ["KNN","LSTM"]
for i in range(len(algorithms)):
    print(""+algorithms[i]+" is: "+str(scores[i])+" %")
sns.set(rc={'figure.figsize':(15,8)})
plt.xlabel("Algorithms")
plt.ylabel(" score")
sns.barplot(algorithms,scores)
plt.show()
import pickle
```

```
pickle.dump(model,open('model.pkl', 'wb'))
```

MAIN:

```
from flask import Flask, render_template, request
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
import pickle
import pandas as pd
from sklearn.model_selection import train_test_split
app = Flask(__name__)
tfvect = TfidfVectorizer(stop_words='english', max_df=0.7)
loaded_model = pickle.load(open('model.pkl', 'rb'))
dataframe = pd.read_csv('news.csv')
x = dataframe['text']
y = dataframe['label']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
def fake_news_det(news):
    tfid_x_train = tfvect.fit_transform(x_train)
    tfid_x_test = tfvect.transform(x_test)
    input_data = [news]
    vectorized_input_data = tfvect.transform(input_data)
    prediction = loaded_model.predict(vectorized_input_data)
    return prediction
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
```

```
    if request.method == 'POST':
        message = request.form['message']
        pred = fake_news_det(message)
        print(pred)
        return render_template('index.html', prediction=pred)
    else:
        return render_template('index.html', prediction="Something went wrong")
if __name__ == '__main__':
    app.run()
```