**Background pattern

Description automatically generated Capstone Project-2**

**Fake News Detection Analysis-LSTM Classification**

**Calendar

Description automatically generated**

PGA-01 Data Analytics and Machine Learning

**(2021-2022)**

**-Submitted by**

**Suganya Janardhanan**

**ABSTRACT:**

Currently, we are in a world of misinformation and fake news. The goal is to detect fake news based on Long Short-Term Memory (LSTM). LSTM networks are an extension of RNN that extend the memory. LSTM are used as the building blocks for the layers of an RNN. LSTM facilitated us to give a sentence as an input for prediction rather than just one word, which is much more convenient in NLP and makes it more efficient. AI/ML-based fake news detector is crucial for companies and media to automatically predict whether circulating news is fake or not.

**Problem statement:**

Develop a deep learning program to identify when a news source may be producing fake news. We aim to use a corpus of labeled real and fake new articles to build a classifier that can make decisions about information based on the content from the corpus. The model will focus on identifying fake news sources, based on multiple articles originating from a source. Once a source is labeled as a producer of fake news, we can predict with high confidence that any future articles from that source will also be fake news. Focusing on sources widens our article misclassification tolerance, because we will have multiple data points coming from each source. The intended application of the project is for use in applying visibility weights in social media. Using weights produced by this model, social networks can make stories which are highly likely to be fake news less visible.

**DATA DEFINITION:**

**id**: unique id for a news article

**title**: the title of a news article

**author**: author of the news article

**text**: the text of the article; could be incomplete

**label**: a label that marks the article as potentially unreliable or reliable

**1**: unreliable **0**: reliable

**TABLE OF CONTENTS:**

**Graphical user interface, text, application, Word

Description automatically generated**

**IMPORT LIBRARIES:**

Imported necessary libraries to structure our code effectively.

**Graphical user interface, text, application

Description automatically generated**

**READ THE DATA:**

Loaded data and displayed first five rows

**Graphical user interface, text, application, Word

Description automatically generated**

Displaying the length of the text and title

Text

Description automatically generated

'House Dem Aide: We Didn’t Even See Comey’s Letter Until Jason Chaffetz Tweeted It By Darrell Lucus on October 30, 2016 Subscribe Jason Chaffetz on the stump in American Fork, Utah (image courtesy Michael Jolley, available under a Creative Commons-BY license) \nWith apologies to Keith Olbermann, there is no doubt who the Worst Person in The World is this week–FBI Director James Comey. But according to a House Democratic aide, it looks like we also know who the second-worst person is as well. It turns out that when Comey sent his now-infamous letter announcing that the FBI was looking into emails that may be related to Hillary Clinton’s email server, the ranking Democrats on the relevant committees didn’t hear about it from Comey. They found out via a tweet from one of the Republican committee chairmen. \nAs we now know, Comey notified the Republican chairmen and Democratic ranking members of the House Intelligence, Judiciary, and Oversight committees that his agency was reviewing emails it had recently discovered in order to see if they contained classified information. Not long after this letter went out, Oversight Committee Chairman Jason Chaffetz set the political world ablaze with this tweet.

From the above text we can see that there are some special characters which we must remove while processing the data.

**CHECKING DATA INFO:**

**Graphical user interface, application

Description automatically generated**

From the information displayed above, we cobserve there are 20800 with 5 respected columns. Also, there are some missing values present in ‘title’, ‘author’, ‘text’. From which the ‘title’, ‘author’, and ‘id’ are not relevant for our the analysis.

**DATA PREPROCESSING:**

**Graphical user interface, text, application, email

Description automatically generated**

Dropped the columns which are not relevant. Also dropping the entire row which contains null vlaues using dropna().

After dropping the unnecessary columns the length of the data is around 20761.

**CONVERTING TO LOWERCASE:**

**Graphical user interface, text, application

Description automatically generated**

Converting a word to lower case (NLP-nlp). Words like Book and book mean the same but, when not converted to the lower case those two are represented as two different words in the vector space model (resulting in more dimensions).

**REPLACING SPECIAL CHARACTERS:**

**Graphical user interface, text, application

Description automatically generated**

As we know, the special characters are non-alphanumeric characters. These characters add no value to text-understanding and induce noise into algorithms.

**STOPWORDS REMOVAL:**

**Graphical user interface, text, application, email

Description automatically generated**

**EXPLORATORY DATA ANALYSIS:**

Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. Graphical user interface, application, Word

Description automatically generated

We observe that the data is balanced.

**FREQUENTLY OCCURRED WORDS IN THE ARTICLE:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Output:**

**Text

Description automatically generated**

**FREQUENT WORDS FOR REAL NEWS:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Output:**

**Text

Description automatically generated**

**FREQUENT WORDS FOR FAKE NEWS:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Output:**

****

**CREATE WORD EMBEDDINGS:**

Before creating word embeddings, we had performed some preprocessing techniques such as tokenization and padding.

**Why tokenization and padding is important?**

Tokenization is the process of splitting a text into list of tokens. Token is an instance of sequence of characters of particular document that are grouped together.

Padding is most important in any text analyzing model. Because, not all the sentences have same length. Some of them will be longer or shorter. We need to have the inputs with same size in lstm. Thus padding helps to bring all the sentence in one range of length.

Graphical user interface, text, application

Description automatically generated

From the above output, we observe there are around 200,000 tokens present in our text.

Graphical user interface, text, application

Description automatically generated

**EMBEDDED MATRIX:**

Creating embedded matrix using ‘GloVe’.

Glove is an algorithm for obtaining vector representation of words.

Graphical user interface, text, application

Description automatically generated

The above output is the vector representation of the embedded words in matrix.

DISPLAYING THE PADDINGS FOR THE INDEX:

Graphical user interface, application, table, Excel

Description automatically generated

**SPLITTING DATA INTO TRAIN AND TEST:**

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(padded\_sequences, data['label'], test\_size=0.20, random\_state=42, stratify=data['label'])

**MODEL BUILDING:**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**MODEL TRAINING:**

history=model.fit(x\_train,y\_train,epochs=10,batch\_size=32,

validation\_data=(x\_test, y\_test))

**The model achieved accuracy of 97% which is excellent.**

**VISUALIZING AND PRINTING THE RESULT:**

**Graphical user interface, application, Word

Description automatically generated**

**Line chart

Description automatically generated with low confidence**

**Graphical user interface, application, Word

Description automatically generated**

**Chart, line chart

Description automatically generated**

**PRINTING FINAL VALUES:**

print(history.history['accuracy'][-1])

print(history.history['val\_accuracy'][-1])

**OUTPUT:**

0.9736873507499695

0.9607512354850769

**CONCLUSION:**

The task of classifying news manually requires in-depth knowledge of the domain and expertise to identify anomalies in the text. In this project, we discussed the problem of classifying fake news articles using deep learning model. The primary aim of the project is to identify patterns in text that differentiate fake articles from true news. We preprocessed the data to extract clean news from the articles using various techniques and algorithm. The learning models were trained and parameter-tuned to obtain optimal accuracy.

Fake news detection has many open issues that require attention of researchers. For instance, to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Thus, Deep learning techniques can be employed to identify the key sources involved in spread of fake news.

Word Cloud is **a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance**