



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)
SRI VENKATESWARA COLLEGE OF ENGINEERING & TECHNOLOGY
(SVCET)

Fake News Detector Using NLP

Final Project By

P. Srinivasan (112421104050)

R. Anand Suman Singh (112421104005)

M. Vikram (112421104057)

J. Hariprasath (112421104019)

Supervisor

Divya

Lecturer, Dept. of CSE

A project submitted in partial fulfilment of the requirements for the degree of
B.E. Engineering in Computer Science and Engineering

Declaration of Authorship

This is to certify that the work that was given has shown the outcome of the analysis and experiments below the superintendence of **Mrs. Divya**,

Lecturer of the Department of applied science and Engineering (CSE), Sri Venkateswara College of Engineering & Technology, Thirupachur, Tiruvallur. It's additionally declared that this project has not been submitted anywhere else for any degree or sheepskin. Info derived from the printed and unpublished work of others has been acknowledged

within the text and a listing of references is given.

Authors

R. Anand Suman Singh

P. Srinivasan

J. Hariprasath

ID : 112421104005

ID : 112421104050

ID : 112421104019

M. Vikram

ID : 112421104057

K. Kiran Kumar

ID : 1124211040

Acknowledgement

First and foremost, praises and thanks to Allah, the Almighty, for His showers of blessings throughout our project work to complete the project successfully.

We would like to express our deep and sincere gratitude to our supervisor, **Mrs.**

Divya, Lecturer, Department of Computer Science & Engineering, Sri Venkateswara College of Engineering & Technology Technology, for allowing us to do the project and providing invaluable guidance throughout this project.

We would be grateful for his guidance. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the project and to present the project works as clearly as possible. It was a great privilege and honour to work and study under his guidance. We are extremely grateful for what he has offered us. We would also like to thank him for his

friendship, empathy, and great sense of humour.

We would also like to thank our parents for being with us during the tough time of pandemic and ensuring that we receive the highest degree.

Abstract

With the recent social media boom, the spread of fake news has become a great concern for everybody. It has been used to manipulate public opinions, influence the election - most notably the US Presidential Election of 2016, incite hatred and riots like the genocide of the Rohingya population. A 2018 MIT study found that fake news spreads six times faster on Twitter than real news. The credibility and trust in the news media are at an all-time low. It is becoming increasingly difficult to determine which news is real and which is fake. Various machine learning methods have been used to separate real news from fake ones. In this study, we tried to accomplish that using Passive Aggressive Classifier, LSTM and natural language processing. There are lots of machine learning models but these two have shown better progress.

Now there is some confusion present in the authenticity of the correctness. But it definitely opens the window for further research. There are some of the aspects that has to be kept in mind considering the fact that fake news detection is not only a simple web interface but also a quite complex thing that includes a lot of backend work.

Table of Content

Declaration of Authorship

Fake News Detector	5
Acknowledgement	3
Abstract	4
Table of Content	4
Introduction	6
2. Problem Statement	7
3. Motivation	8
4. Background Study	11
5. Feasibility Study	13
6. Methodology	14
6.1 The Dataset	14
6.2 The Machine Learning Model	15
6.3 The Web Interface	17
6.4 Common Platform: Flask	18
7. Implementation	18
7.1 The Interface	18
7.2 The ML Model	19
7.3 Flask Code	33
7.4 Web Interface	35
8. Key Insights	39
9. Conclusion	39
10. Future Work	39
11. References	41

1. Introduction

Fake news is untrue information presented as news. It often has the aim of damaging the reputation of a person or entity or making money through advertising revenue. Once common in print, the prevalence of fake news has increased with the rise of social media, especially the Facebook News Feed. During the 2016 US presidential election, various kinds of fake news about the candidates widely spread in the online social networks, which may have a significant effect on the election results. According to a post-election statistical report, online social networks account for more than 41.8% of the fake news data traffic in the election, which is much greater than the data traffic shares of both traditional TV/radio/print medium and online search engines respectively. Fake news detection is becoming increasingly difficult because people who have ill intentions are writing the fake pieces so convincingly that it is difficult to separate from real news. What we have done is a simplistic approach that looks at the news headlines and tries to predict whether they may be fake or not.

Fake news can be intimidating as they attract more audience than normal. People use them because this can be a very good marketing strategy. But the money earned might not live upto fact that it can harm people.

2. Problem Statement

In this day and age, it is extremely difficult to decide whether the news we come across is real or not. There are very few options to check the authenticity and all of them are sophisticated and not accessible to the average person. There is an acute need for a web-based fact-checking platform that harnesses the power of Machine Learning to provide us with that opportunity.

3. Motivation

Social media facilitates the creation and sharing of information that uses computer-mediated technologies. This media changed the way groups of people interact and communicate. It allows low cost, simple access and fast dissemination of information to them. The majority of people search and consume news from social media rather than traditional news organizations these days. On one side, where social media have become a powerful source of information and bringing people together, on the other side it also put a negative impact on society. Look at some examples herewith; Facebook Inc's popular messaging service, WhatsApp became a political battle-platform in Brazil's election. False rumours, manipulated photos, de-contextualized videos, and audio jokes were used for campaigning. These kinds of stuff went viral on the digital platform without monitoring their origin or reach. A nationwide block on major social media and messaging sites including Facebook and Instagram was done in Sri Lanka after multiple terrorist attacks in the year 2019. The government claimed that "false news reports" were circulating online. This is evident in the challenges the world's most powerful tech companies face in reducing the spread of misinformation. Such examples show that Social Media enables the widespread use of "fake news" as well. The news disseminated on social media platforms may be of low quality carrying misleading information intentionally. This

sacrifices the credibility of the information. Millions of news articles are being circulated every day on the Internet – how one can trust which is real and which is fake? Thus incredible or fake news is one of the biggest challenges in our digitally connected world. Fake news detection on social media has recently become an emerging research domain. The domain focuses on dealing with the sensitive issue of preventing the spread of fake news on social media. Fake news identification on social media faces several challenges. Firstly, it is difficult to collect fake news data. Furthermore, it is difficult to label fake news manually. Since they are intentionally written to mislead readers, it is difficult to detect them simply based on news content. Furthermore, Facebook, Whatsapp, and Twitter are closed messaging apps. The misinformation disseminated by trusted news outlets or their friends and family is therefore difficult to be considered as fake. It is not easy to verify the credibility of newly emerging and time-bound news as they are not sufficient to train the application dataset. Significant approaches to differentiate credible users, extract useful news features and develop authentic information dissemination systems are some useful domains of research and need further investigations. If we can't control the spread of fake news, the trust in the system will collapse. There will be widespread distrust among people. There will be nothing left that can be objectively used. It means the destruction of political and social coherence. We wanted to build some sort of web-based system that can fight this nightmare scenario. And we made

Fake News Detector
some significant progress towards that goal.

10

4. Background Study

From an NLP perspective, researchers have studied numerous aspects of the credibility of online information. For example, [1] applied the time-sensitive supervised approach by relying on tweet content to address the credibility of a tweet in different situations. [2] used LSTM in a similar problem of early rumour detection. In another work, [3] aimed at detecting the stance of tweets and determining the veracity of the given rumour with convolution neural networks. A submission [4] to the SemEval 2016 Twitter Stance Detection task focuses on creating a bag-of-words autoencoder and training it over the tokenized tweets. Another team, [5], combined multiple models in an ensemble providing a 50/50 weighted average between a deep convolutional neural network and a gradient-boosted decision tree. Though this work seems to be similar to our work, the difference lies in the construction of an ensemble of classifiers. In a similar attempt, a team [6] concatenated various features vectors and passed them through an NLP model. Passive Aggressive algorithm is a margin-based online learning algorithm for binary classification. It is also an algorithm of a soft margin-based method and robust to noise. It can be used in fake news detection [16] Term Frequency-Inverse Document Frequency is also a method used to represent text in a format that can be easily processed by machine learning algorithms. It is a numerical

statistic that shows how important a word is to news in a news dataset. The importance of a word is proportional to the number of times the word appears in the news (fake and real) but inversely proportional to the number of times the word appears in the news dataset (fake or real) [15]

5. Feasibility Study

Passive-aggressive classifier, logistic regression, LSTM can be used in fake news detection. Bi-directional LSTM was used in [7] to detect fake news. It had reasonably good accuracy but if the news was a bit more sophisticated, it would be difficult to achieve good accuracy. Because this model picks up the sensational/clickbaity words as part of fake news. For example, if a news title says, ‘Donald Trump is the greatest president ever, the model will pick it up as fake news with reasonable accuracy. If the title is more nuanced and written in a sophisticated way, it’d be difficult to do so. We believe that our LSTM model is not enough by itself to detect fake news. That’s why we included passive aggressive classifier with it and when we compared passive news with reputable news sources, but the scope of the work is so vast that we couldn’t do it with the resources available to us. Our model can act as a first step in detecting fake news.

But more work is needed to call the model reliable enough.

6. Methodology

6.1 The Dataset

The screenshot shows a Microsoft Excel spreadsheet titled "news - Excel". The data is organized into four columns: "title", "text", "C", and "label". The "label" column contains the classification of each news item as either "FAKE" or "REAL". The first few rows of data are as follows:

title	text	C	label
2. You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fellow at the FAKE		FAKE
3. Watch The Exact Moment Paul Ryan Committed Politic Google Pinterest Digg LinkedIn Reddit Stumbleupon	U.S. Secretary of State John F. Kerry said Monday that		REAL
4. Kerry to go to Paris in gesture of sympathy	Bernie supporters on Twitter erupt in anger against th� Kaydee King (@KaydeeKing) November 9, 2016		FAKE
5. Bernie supporters on Twitter erupt in anger against th� Kaydee King (@KaydeeKing) November 9, 2016	It's primary day in New York and front-runners Hillary		REAL
6. The Battle of New York: Why This Primary Matters	Tehran, USA		FAKE
7. Tehran, USA	Girl Horrified At What She Watches Boyfriend Do After Share This Baylee Luciani (left), Screenshot of what		FAKE
8. Girl Horrified At What She Watches Boyfriend Do After Share This Baylee Luciani (left), Screenshot of what	A Czech stockbroker who saved more than 650 Jewish c		REAL
9. � Britainâ€™s Schindlerâ€™ Dies at 106	Fact check: Trump and Clinton at the 'Commander-in-cl Hillary Clinton and Donald Trump made some		REAL
10. Fact check: Trump and Clinton at the 'Commander-in-cl Hillary Clinton and Donald Trump made some	Iran reportedly makes new push for uranium concessions Iranian negotiators reportedly have made a last-ditch		REAL
11. Iran reportedly makes new push for uranium concessions Iranian negotiators reportedly have made a last-ditch	With all three Clintons in Iowa, a glimpse at the fire th CEDAR RAPIDS, Iowa ��el had one of the most		REAL
12. With all three Clintons in Iowa, a glimpse at the fire th CEDAR RAPIDS, Iowa ��el had one of the most	Donald Trumpâ€™s Shockingly Weak Delegate Game S Donald Trumpâ€™s organizational problems have		REAL
13. Donald Trumpâ€™s Shockingly Weak Delegate Game S Donald Trumpâ€™s organizational problems have	Strong Solar Storm, Tech Risks Today SO News Oct.26 Click Here To Learn More About Alexandra's		FAKE
14. Strong Solar Storm, Tech Risks Today SO News Oct.26 Click Here To Learn More About Alexandra's	10 Ways America Is Preparing for World War 3		FAKE
15. 10 Ways America Is Preparing for World War 3	October 31, 2016 at 4:52 am		FAKE
16. Trump takes on Cruz, but lightly	Killing Obama administration rules, dismantling Obam		REAL
17. How women lead differently	As more women move into high offices, they often		REAL
18. Shocking! Michele Obama & Hillary Caught Glamorizin Shocking! Michele Obama & Hillary Caught	Shocking! Michele Obama & Hillary Caught		FAKE
19. Hillary Clinton in HUGE Trouble After America Noticed O	Hillary Clinton in HUGE Trouble After America Noticed O		FAKE
20. What's in that Iran bill that Obama doesn't like?	What's in that Iran bill that Obama doesn't like? Washington (CNN) For months, the White House and		REAL
21. What's in that Iran bill that Obama doesn't like?	The 1 chart that explains everything you need to know While paging through Pew's best data visualizations of		REAL
22. The slippery slope to Trumpâ€™s proposed ban on Mu With little fanfare this fall, the New York developer	The slippery slope to Trumpâ€™s proposed ban on Mu With little fanfare this fall, the New York developer		REAL
23. Episode #160 �� SUNDAY WIRE: �� Hail to the Deplora November 13, 2016 By 21Wire Leave a Comment	Episode #160 �� SUNDAY WIRE: �� Hail to the Deplora November 13, 2016 By 21Wire Leave a Comment		FAKE
24. Hillary Clinton Makes A Bipartisan Appeal on Staten Isl Hillary Clinton told a Staten Island crowd today that	Hillary Clinton Makes A Bipartisan Appeal on Staten Isl Hillary Clinton told a Staten Island crowd today that		REAL
25. New Senate majority leaderâ€™s main goal for GOP: D Mitch McConnell has an unusual admonition for the	New Senate majority leaderâ€™s main goal for GOP: D Mitch McConnell has an unusual admonition for the		REAL

Figure 1 : Dataset

The dataset is simple. It contains the titles of the news, the body text and a label field, which, if the news is authentic, shows REAL and if inauthentic, shows FAKE.

There are 3 main segments of the methodology :

- The core Machine Learning model.
- The web interface.
- The common platform that brings the model and the interface together.

6.2 The Machine Learning Model

There are two parts to the ML Model building. Machine Learning is a part of our life that can help us in predicting. We are using two types of model in this case. For the first part, we used passive-aggressive classifiers. And the steps include:

1. **Data Loading:** We are loading a CSV file for the data sorting and training-testing part of the model. The CSV file is turned into an array for easier work purpose.
2. **Vectorization:** Vectorization is needed for determining the frequency of the words present in a passage. This is needed to determine which words are used often.
3. **Classifier:** Passive-aggressive algorithms are a family of great learning algorithms. They are similar to Perceptron because it does not require a reading scale. However, unlike Perceptron, they include parameter correction. Passive is used when the prediction is correct and there is no change in the model. But if there is any kind of change in the model, that is if the prediction is not correct then the aggressive part is called, which changes the model accordingly. The aggressive part of the model changes

the model according to its wish on the backend.

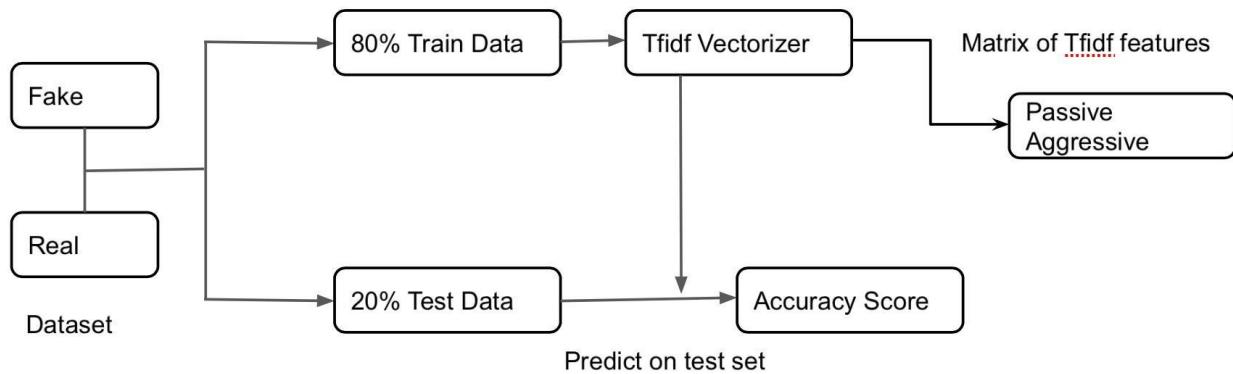


Figure 2 : Passive-aggressive model

4. Model Building: The model is built through the train and test of the dataset, by ensuring that the training is done for 80% of the dataset and testing is done in the rest of the 20% of the dataset.

In the second part, we used is LSTM. Here are the steps :

1. **Loading the data:** For this step, it is the same as the passive-aggressive one.
2. **Scanning and parsing.** Data is loaded from a CSV file. This consists of the body of selected news articles. It then contains a label field that indicates whether the news is real or fake. In this code block, we scan the CSV and clean the titles to filter out stop words and punctuation.

3. Tokenization. The tokenizer is used to assign indices to words, and filter out infrequent words. This allows us to generate sequences for our training and testing data.

4. Embedding matrix: Apply the embedding matrix. An embedding matrix is used to extract the semantic information from the words in each title.

5. Model Building: Building the model and finding out the accuracy via confusion matrix. The model is created using an Embedding layer, LSTM, Dropout, and Dense layers. We are going to run the data on 20 epochs. We observed that the LSTM model is vastly inaccurate in predicting the authenticity of the news. So we decided to show the output by running it through the Passive-aggressive classifier model.

6.3 The Web Interface

This was the simplest part.

1. **HTML for building the basic skeleton:** HTML makes the structure of the web application and also there are some of the functions that can be achieved best with HTML only.
2. **CSS for design:** The CSS part is for designing only. Because it will give a more beautiful aspect to the website.

6.4 Common Platform: Flask

This acts as a common platform and takes the input with the pickle module and passes it to the machine learning model afterwards the prediction is shown on the screen with the HTML and CSS website.

1. Building functions for taking input.
2. Passing input values through the ML model.
3. Using the Pickle module for serializing and de-serializing the dataset.
4. Providing output.

7. Implementation

7.1 The Interface

This is what you see when you go to the web interface. You are supposed to copy the news and paste it into the input box.

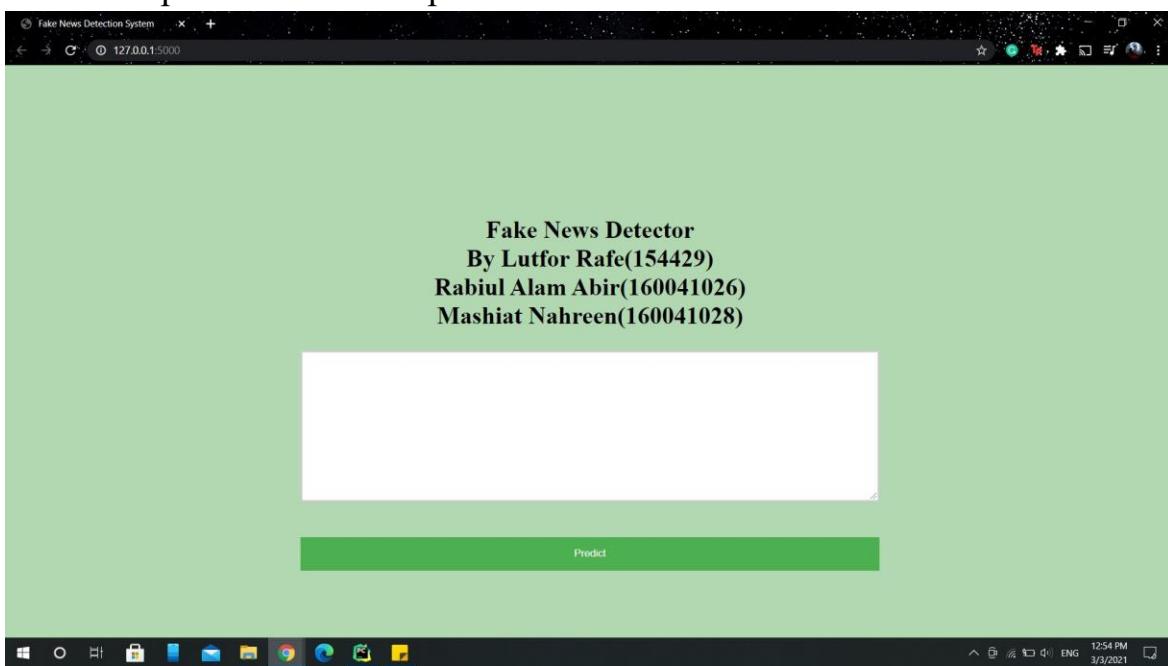


Figure 3.1 : The Interface

When you paste the news on the input box and click ‘Predict’ the model will give you the result. If the news seems authentic, the output will be ‘Looking Real News’. Otherwise, it will show ‘Looking Fake News’. That’s how you can detect fake or real news via the interface.

**Figure 3.2 : The Interface**

7.2 The ML Model

The code for the ML model building is as follows:

TF-IDF stands for Term Frequency-Inverse Document Frequency. Term frequency is basically a ratio of the number of times a particular word appears with respect to the total number of words. And Inverse Document Frequency is basically the weight

```
from sklearn.feature_extraction.text import TfidfVectorizer
text = ['This is the final project of Mashiat Nahreen, Lutfor Rafe and Rabiul Alam Abir', 'This is the final project of our undergrad.']
vectorization = TfidfVectorizer().fit(text)
print(vectorization.idf_)
print(vectorization.vocabulary_)
```

Words that are present in every data will have very low IDF value and using that we will highlight the maximum IDF values.

```
example = text[0]
example =
vectorization.transform([example])
print(example.toarray())
```

The zeros represent there are no words in that position.

IMPLEMENTING PASSIVE AGGRESSIVE CLASSIFIER

Passive is used when the prediction is correct and there is no change in the model. But if there is any kind of change in the model that is if the prediction is not correct then aggressive part is called, which changes the model accordingly.

```
import os
os.chdir("D:\Books\Fake_News_Detection-master")
```

OS module is used for the Python program to interact with the operating system

```
import pandas as pd dataset =  
  
pd.read_csv('news.csv') dataset.head() x =  
  
dataset['text'] y = dataset['label'] x y from  
  
sklearn.model_selection import train_test_split  
  
from         sklearn.feature_extraction.text      import  
  
TfidfVectorizer  
  
from         sklearn.linear_model                import  
PassiveAggressiveClassifier  
  
from         sklearn.metrics                   import  
accuracy_score,  
  
confusion_matrix  
  
x_train,x_test,y_train,y_test =  
  
train_test_split(x,y,test_size=0.2,random_state=0)  
  
y_train y_train  
  
vectorization =  
  
TfidfVectorizer(stop_words='english',max_df=0.7)  
  
xv_train = vectorization.fit_transform(x_train)  
  
xv_test = vectorization.transform(x_test)
```

max_df refers to the percentage of the repetition of the word. 0.7 means 70% of the time the word is repeated.

```
classifier = PassiveAggressiveClassifier(max_iter=50)

classifier.fit(xv_train, y_train) y_pred =

classifier.predict(xv_test)

score = accuracy_score(y_test, y_pred) print(f'Accuracy:

{round(score*100, 2)}%')

cf = confusion_matrix(y_test, y_pred,

labels=['FAKE', 'REAL'])

print(cf) def

fake_news_det(news):

    input_data = [news]
        vectorized_input_data =
vectorization.transform(input_data)
            prediction =
classifier.predict(vectorized_input_data)

print(prediction) fake_news_det('U.S. Secretary of State

John F. Kerry said Monday that he will stop in Paris

later this week, amid criticism that no top American

officials

attended Sunday's unity march against terrorism.')

fake_news_det("""Go to Article
```

President Barack Obama has been campaigning hard for the woman who is supposedly going to extend his legacy four more years. The only problem with stumping for Hillary Clinton, however, is she's not exactly a candidate easy to get too enthused about.

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

pickle is used for serializing and deserializing any data that is inputted in Python.

```
loaded_model = pickle.load(open('model.pkl','rb')) def fake_news_det1(news):
    input_data = [news]
    vectorized_input_data = vectorization.transform(input_data)
    prediction = classifier.predict(vectorized_input_data)
    print(prediction) fake_news_det1("""U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this week, amid criticism that no top American officials attended Sunday's unity march against
```

```
terrorism."") fake_news_det('''U.S. Secretary of State  
John F. Kerry said Monday that he will stop in Paris  
later this week, amid criticism that no top American  
officials  
attended Sunday's unity march against terrorism.'')
```

In this project, titles of news articles found on the internet is used to determine whether a news is fake or real. We are using LSTM to help classify them into either real or fake category.

```
import numpy as np import pandas as pd import json  
as j import urllib import gzip import nltk  
nltk.download('stopwords') from nltk.stem import  
PorterStemmer from sklearn.model_selection import  
train_test_split  
  
!pip install gensim  
from gensim.models import KeyedVectors  
  
from nltk.corpus import stopwords from  
keras.models import Model  
  
from keras.callbacks import EarlyStopping,  
ModelCheckpoint  
  
from keras.layers import Dense, Input, LSTM,
```

Embedding, Dropout, Activation `from`

```
keras.layers.merge import concatenate  
from keras.layers.normalization import  
BatchNormalization from keras.preprocessing import  
sequence from keras.preprocessing.text import  
Tokenizer from keras.preprocessing.sequence import  
pad_sequences
```

Data scanning and parsing : Data is loaded from a csv file `fake_or_real_news.csv`.

This consists of the title and text of a select group of news articles. It then contains a label field which indicates whether the news is real or fake. In this code block, we scan the csv and clean the titles to filter out stop words and punctuation.

```
import re  
import string  
from  
sklearn.feature  
_extraction.tex  
t import
```

```
CountVectorizer def

clean_text(text):

text = str(text)

text = text.split()

words = [] for word

in text:

exclude = set(string.punctuation)

word = ''.join(ch for ch in word if ch not in

exclude) if word in

stops:

continue

try:

words.append(ps.stem(word))

except UnicodeDecodeError:

words.append(word)

text = " ".join(words)

return text.lower() stops

= set(stopwords.words("engli
```

```
sh")) ps = PorterStemmer()

f =
pd.read_csv('news.csv')

f.label = f.label.map(dict(REAL=1, FAKE=0))

f
```

We take the news titles and divide the train and test set. We also clean the text.

```
f = f[1:100]

x_train, x_test, y_train, y_test =
train_test_split(f['title'], f.label, test_size=0.2)

X_cleaned_train = [clean_text(x) for x in X_train]

X_cleaned_test = [clean_text(x) for x in X_test]

X_cleaned_train[0]
```

Tokenizer : Tokenizer is used to assign indices to words, and filter out infrequent words. This allows us to generate sequences for our training and testing data.

```
import tokenize from keras.preprocessing.text

import Tokenizer MAX_NB_WORDS = 20000

tokenizer = Tokenizer(num_words=MAX_NB_WORDS)

tokenizer.fit_on_texts(X_cleaned_train + X_cleaned_test) print('Finished
Building Tokenizer')
```

```
train_sequence =  
tokenizer.texts_to_sequences(X_cleaned_train)  
print('Finished Tokenizing Training')
```

```
test_sequence =  
tokenizer.texts_to_sequences(X_cleaned_test)  
  
print('Finished Tokenizing Training')
```

Embedding Matrix : Embedding matrix is used to extract the semantic information from the words in each title.

```
from gensim.models import KeyedVectors  
  
from gensim.models import Word2Vec  
  
EMBEDDING_FILE =  
  
'https://s3.amazonaws.com/dl4j-distribution/GoogleNews  
-vectors-negative300.bin.gz'  
  
Word2Vec =  
  
KeyedVectors.load_word2vec_format(EMBEDDING_FILE,  
binary=True)  
  
word_index = tokenizer.word_index print('Found %s  
unique tokens' % len(word_index)) nb_words =  
min(20000, len(word_index)) embedding_matrix =  
np.zeros((nb_words, 300)) for word, i in  
word_index.items():
```

```
try:  
  
    embedding_vector = word2vec.word_vec(word) if  
  
    embedding_vector is not None and i < 7000:  
  
        embedding_matrix[i] = embedding_vector  
  
except (KeyError, IndexError) as e:  
  
    continue
```

Building the Model : The model is created using an Embedding layer, LSTM,

Dropout, and Dense layers. We are going to run the data on 20 epochs.

```
from keras.models import Sequential from keras.layers  
import Dense, LSTM, Dropout, Conv1D, MaxPooling1D from  
keras.layers.embeddings import Embedding from  
keras.preprocessing import sequence from  
keras.preprocessing.sequence import pad_sequences  
kVECTORLEN = 50 model = Sequential()  
model.add(Embedding(5000, 500, input_length=50))  
model.add(Dropout(0.4)) model.add(Dense(1,  
activation='relu'))  
model.compile(loss='binary_crossentropy',  
optimizer='adam', metrics=['accuracy'])  
print(model.summary())
```

```
Fake News Detector                                     30
train_sequence = sequence.pad_sequences(train_sequence,          maxlen=50)

test_sequence = sequence.pad_sequences(test_sequence,
maxlen=50)

history = model.fit(train_sequence, y_train,
validation_data=(test_sequence, y_test), epochs=20,
batch_size=64)
```

Calculating the accuracy.

```
scores = model.evaluate(test_sequence, y_test,
verbose=0)
accuracy = (scores[1]*100) print("Accuracy:
{:.2f}%".format(scores[1]*100))
```

Analyzing the Data: The graphs below demonstrate the change in accuracy and loss for the training data as well as the validation data.

```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'])

plt.plot(history.history['val_accuracy'])

plt.title('model accuracy') plt.ylabel('accuracy')

plt.xlabel('epoch') plt.legend(['train',
'validation'], loc='upper left') plt.show()

plt.plot(history.history['loss'])
```

```
plt.plot(history.history['val_loss'])

plt.title('model loss') plt.ylabel('loss')

plt.xlabel('epoch') plt.legend(['train', 'test'],
loc='upper left') plt.show()
```


7.3 Flask Code

```
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__) vectorization =
TfidfVectorizer(stop_words='english', max_df=0.7)
loaded_model = pickle.load(open('model.pkl', 'rb'))
dataset = pd.read_csv('news.csv') x = dataset['text'] y
= dataset['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):
    xv_train = vectorization.fit_transform(x_train)
    xv_test = vectorization.transform(x_test)
    input_data = [news]
    vectorized_input_data =
vectorization.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(debug=True)
```

7.4 Web Interface

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Fake News Detection System</title>

<link

href='https://fonts.googleapis.com/css?family=Pacifico'

' rel='stylesheet' type='text/css'>

<link

href='https://fonts.googleapis.com/css?family=Arimo'

rel='stylesheet' type='text/css'>

<link

href='https://fonts.googleapis.com/css?family=Hind:300

' rel='stylesheet' type='text/css'>

<link

href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<meta name="viewport" content="width=device-width,
initial-scale=1">
```

```
<style> input[type=text], select,  
textarea { width: 50%; padding:  
10px; border: 3px solid #ccc;  
border-radius: 1px; box-sizing:  
border-box; margin-top: 6px; margin-  
bottom: 16px; resize: horizontal;  
}  
button { background-color:  
#4CAF50; color: white; padding:  
14px 20px; margin: 8px 0; border:  
none;  
cursor: pointer;  
width: 50%; }  
  
button:hover {  
opacity: 0.8;  
}  
  
h1 {  
text-align: center;  
}
```

```
    p {
        text-align: center;
    }

    div {
        text-align: center;
    } body {
        background: rgba(0, 128, 0, 0.3) /* Green
background with 30% opacity */
    }
}

</style>

</head>

<body>

<p style="padding: 0 10em 10em 0">

<div class="login">
    <h1 style="text-align:center;">Fake News Detector
    <br> By Lutfor Rafe(154429) <br> Rabiul Alam
    Abir(160041026) <br> Mashiat
    Nahreen(160041028) </h1>

    <form action="{{ url_for('predict') }}"
method="POST">
```

```
<textarea name="message" rows="6" cols="20"

required="required" style="font-size:

18pt"></textarea>

<br> <br>

<button type="submit" class="btn btn-primary

btn-block btn-large">Predict</button>

<div class="results">

{ % if prediction == ['FAKE'] %}

<h2 style="color:red;">Looking SpamNews </h2>

{ % elif prediction == ['REAL'] %}

<h2 style="color:green;"><b>Looking Real

News 
```

8. Key Insights

The passive aggressive model produces 93% accuracy. When we input the news text on the interface, it correctly identifies the news most of the time. We tested this by using news from The Onion. The Onion is a satire ‘news’ portal that posts fake funny news. When we pasted some of the news from the site on our web interface, those were correctly identified as fake. But when we wanted to test the news from BBC or New York Times, those were correctly identified as real. But the accuracy of the LSTM model was much lower, so we went with the Passive Aggressive model to produce output on the interface.

9. Conclusion

Our project can ring the initial alert for fake news. The model produces worse results if the article is written cleverly, without any sensationalization. This is a very complex problem but we tried to address it as much as we could. We believe the interface provides an easier way for the average person to check the authenticity of a news. Projects like this one with more advanced features should be integrated on social media to prevent the spread of fake news.

10. Future Work

There are many future improvement aspects of this project. Introducing a cross checking feature on the machine learning model so it compares the news inputs with

the reputable news sources is one way to go. It has to be online and done in real time, which will be very challenging. Improving the model accuracy using bigger and better datasets, integrating different machine learning algorithms is also something we hope to do in the future.

11. References

- [1] C. Castillo, M. Mendoza, and B. Poblete. Predicting information credibility in time-sensitive social media. *Internet Research*, 23(5):560–588, 2013.
- [2] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang. Call attention to rumours: Deep attention-based recurrent neural networks for early rumour detection. *arXiv preprint arXiv:1704.05973*, 2017.
- [3] Y.-C. Chen, Z.-Y. Liu, and H.-Y. Kao. Ikm at several-2017 task 8: Convolutional neural networks for stance detection and rumour verification. *Proceedings of SemEval. ACL*, 2017.
- [4] I. Augenstein, A. Vlachos, and K. Bontcheva. Usfd at semeval-2016 task 6: Any-target stance detection on Twitter with autoencoders. In *SemEval@NAACL-HLT*, pages 389–393, 2016.
- [5] S. B. Yuxi Pan, Doug Sibley. Talos. <http://blog.talosintelligence.com/2017/06/>, 2017.
- [6] B. S. Andreas Hanselowski, Avinesh PVS and F. Caspelherr. Team athene on the fake news challenge. 2017.
- [7] Bahad, P., Saxena, P. and Kamal, R., 2019. Fake News Detection using Bi-directional LSTM-Recurrent Neural Network. *Procedia Computer Science*, 165, pp.74-82.
- [8] EANN: Event Adversarial Neural Networks for Multi-Modal

[9] Fake News Detection on Social Media: A Data Mining Perspective Kai Shuy,

Amy Slivaz, Suhang Wangy, Jiliang Tang \, and Huan Liuy

[10] CSI: A Hybrid Deep Model for Fake News DetectionIdentifying the signs of

fraudulent accounts using data mining techniques Shing-Han Li a,, David C. Yen

b,1, Wen-Hui Luc,2, Chiang Wanga,2

[11] Automatic Deception Detection: Methods for Finding Fake News. Niall J.

Conroy, Victoria L. Rubin, and Yimin Chen

[15] J. D'Souza, "An Introduction to Bag-of-Words in NLP," 03 04 2018. [Online].

Available:<https://medium.com/greyatom/an-introduction-tobag-of-words-in-nlp-ac>

967d43b428.

[16] G. Bonacorso, "Artificial Intelligence – Machine Learning – Data Science,"

10 06 2017.

[Online]. Available:<https://www.bonacorso.eu/2017/10/06/mlalgorithms-addendum-passive-aggressivealgorithms/>