ADVANCED ACADEMIC CENTER

A CENTER FOR INTER-DISCIPLINARY RESEARCH
2021-22

TITLE

“FAKE NEWS DETECTION”

SUPERVISED BY

ROHITH REDDY

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
AUTONOMOUS

Advanced Academic Center

(A Center For Inter-Disciplinary Research)

This is to certify that the project titled

“FAKE NEWS DETECTION”

is a bonafide work carried out by the following students in partial fulfilment of the requirements for Advanced Academic Center intern, submitted to the chair, AAC during the academic year 2020-21.

NAME	ROLL NO.	BRANCH
ROHITHA TUNKIPATI	21241A6662	CSM
RITIKA KOSIGI SHROFF	21241A05F6	CSE
KOLANI ANUP REDDY	21241A05U5	CSE

NAME	ROLL N0	BRANCH
ROHIT KIRAN ELTEM	21241A1254	IT

This work was not submitted or published earlier for anystudy

Dr/Ms./Mr.

Project Supervisor

Dr.B.R.K.Reddy
Program Coordinator

Dr.Ramamurthy Suri
Associate Dean,AAC

ADVANCED ACADEMIC CENTER

ACKNOWLEDGEMENTS

We express our deep sense of gratitude to our respected Director, Gokaraju Rangaraju Institute of Engineering and Technology, for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we extend our appreciation to our respected Principal, for permitting us to carry out this project.

We are thankful to the Associate Dean, Advanced Academic Centre, for providing us an appropriate environment required for the project completion.

We are grateful to our project supervisor who spared valuable time to influence us with their novel insights.

We are indebted to all the above mentioned people without whom we would not have concluded the project.

FAKE NEWS DETECTION

ABSTRACT

- Intentionally deceptive content presented under the guise of legitimate journalism is a worldwide information accuracy and integrity problem that affects opinion forming, decision making, and voting patterns. Most so called "fake news" is initially distributed over social media conduits like Facebook and Twitter and later finds its way onto mainstream platforms such as traditional television and radio news. The fake news stories that are initially seeded over social media platforms share key linguistic characteristics such as making excessive use of unsubstantiated hyperbole and non attributed quoted content.

- In this project, the results of a fake news identification study that documents the performance of a fake news classifier are presented

INTRODUCTION

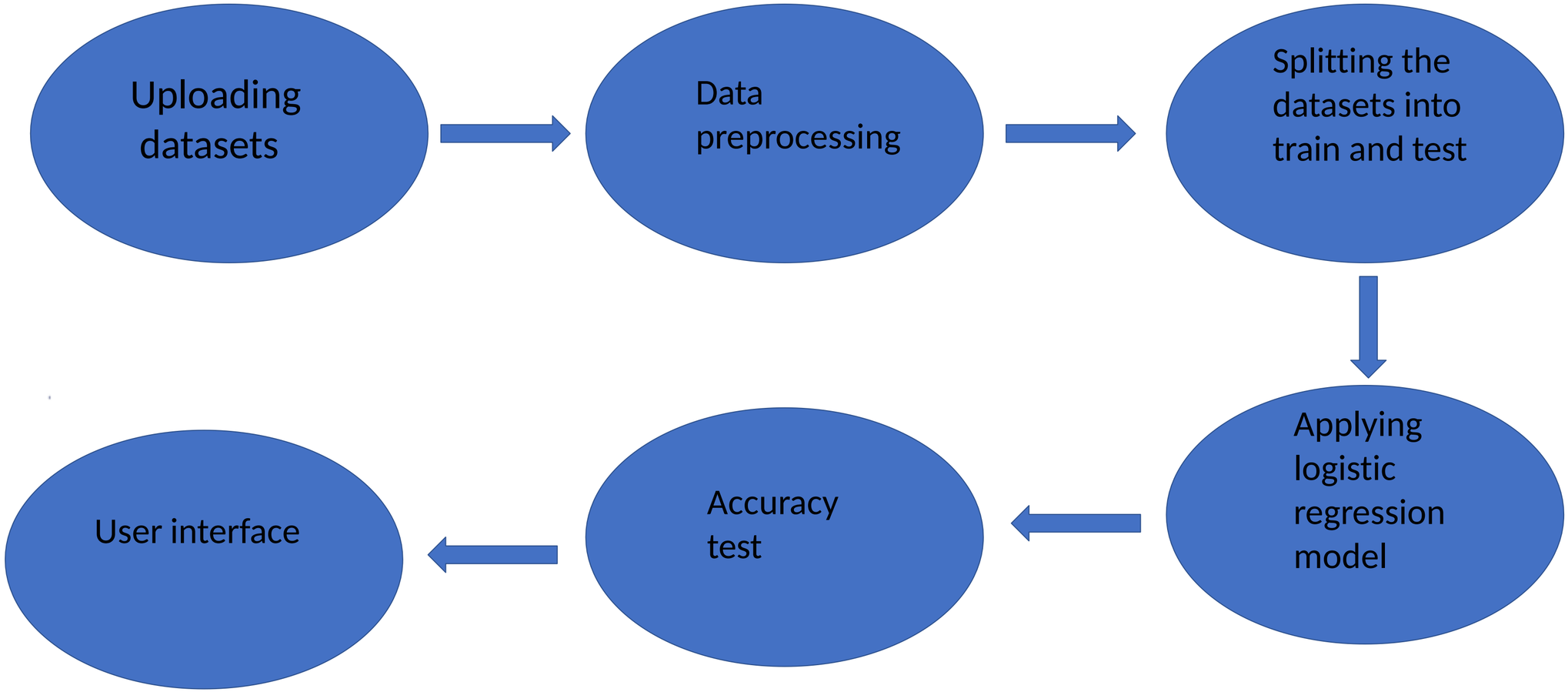
- Intentionally deceptive content presented under the guise of legitimate journalism is a worldwide information accuracy and integrity problem that affects opinion forming, decision making, and voting patterns. Most so called "fake news" is initially distributed over social media conduits like Facebook and Twitter and later finds its way onto mainstream platforms such as traditional television and radio news. The fake news stories that are initially seeded over social media platforms share key linguistic characteristics such as making excessive use of unsubstantiated hyperbole and non attributed quoted content.The results of a fake news identification study that documents the performance of a fake news classifier are presented and discussed in this paper.

ALOGRITHM

- Logistic regression:Logistic regression is a class of regression where the independent variable is used to predict the dependent variable. When the dependent variable has two categories, then it is a binary logistic regression.
- This statistical model is mostly used for classification and predictive systems.It estimates the probability of event occurance.
- Natural language preprocessing(NLP):It is the ability of the machine to read,write,understand and derive meaning from a human language.
- At first, Data preprocessing can be done which helps to clean the data later it follows some methods like.

1.Tokenisation- splitting each word of a sentence as a token
2.Stemming
3.Lemmatization
4.Parts of speech(POS)tagging-It is generally referred as parsing. It recognises the parts of speech of each token
5.Named entity recognition-it involves identification of key information in text and classification into a predefined categoriesEg:Sunder Pichai is CEO of google
6.Chunking-As the name says, it means breaking the text into chunks for better working. Here we have used 1)Stop words removal:These are commonly used words and are removed from the text as they do not add any value
.2)stemming:This is text standarisation step where the words are stemmed or diminished to their roots/base form.
3)Lemmatization:It has pre-defined dictionary that stores the context of words and checks the word in the dictionary while diminishing.

Project Workflow



Data preprocessing:

- ➢ Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
- ➢ When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task

- ➢ Data preprocessing includes the following steps

- Downloading stop words
- Replacing null values with empty strings
- Stemming (reducing a word to its root word)
- Vectorization (converting text format to numerical format)

➢ Splitting into training and test data

- The train-test split is used to estimate the performance of machine learning algorithms that are applicable for prediction-based Algorithms/Applications. This method is a fast and easy procedure to perform such that we can compare our own machine learning model results to machine results.
- Test set is split into 20 % of actual data and the training set is split into 80% of the actual data.

➢Logistic regression model

- It is a Machine Learning classification algorithm that is used to predict the probability of a categorical
  dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc) or 0 (no, failure, etc.).
  In other words, the logistic regression model predicts P(Y=1) as a function of X.

➢ Accuracy test

- Accuracy is a metric used in classification problems used to tell the percentage of accurate predictions. We calculate it by dividing the number of correct predictions by the total number of predictions. This formula provides an easy-to-understand definition that assumes a binary classification problem

- Accuracy score of the training data : 0.98359375

- Accuracy score of the test data : 0.9591346153846154

➢ User interface

- To display input n output, we created an interface(GUI). That's build up with python library namely tkinter. If we give an input on the top box, we get output on dialogue box. Attributes are taken according to our choice. As our code is in python, we created a GUI with python library rather than preferring html, CSS or some other languages.

Code

In [1]:
```
import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [2]:
```
import nltk
nltk.download('stopwords')
```

In [3]:
```
dataset=pd.read_csv('train.csv')
```

In [4]:
```
dataset
```

In [5]:
```
print(stopwords.words('english'))
```

In [6]:
```
dataset.head()
```

In [7]:
```
#missing values
dataset.isnull().sum()
```

In [8]:
```
#replacing null values with empty string ('' represents empty string)
dataset=dataset.fillna('')
```

In [9]:
```
# merging authors name and newss title
dataset['content']=dataset['text']+''+dataset['title']
```

In [10]:
```
dataset['content']
```

In [11]:
```
#separating the data and labels
X = dataset.drop(columns='label', axis=1)
Y = dataset['label']
```

In [12]:
```
X
Y
```

In [13]:
```
#stemmimg ==>reducing a word to its root word
port_stem=PorterStemmer()
```

In [14]:
```
import string
import re
from nltk.corpus import stopwords
stop_word = stopwords.words('english')
def cleaning_data(x):
    x=x.lower()
    x = ' '.join([word for word in x.split(' ') if word not in stop_word])
    return x
dataset.text=dataset.text.apply(cleaning_data)
```

In [15]:
```
dataset['content'] = dataset['content'].apply(cleaning_data)
```

In [16]:
```
dataset['content']
```

In [17]:
```
#separating data and label
X=dataset['content'].values
Y=dataset['label'].values
```

In [18]:
```
X
```

In [19]:
```
Y
```

In [20]:
```
#converting textual data to numerical data using tfidf vectorizer
vectorizer = TfidfVectorizer()
vectorizer.fit(X)

X=vectorizer.transform(X)
```

In [21]:
```
print(X)
```

In [22]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2)
```

In [23]:
```
model  =LogisticRegression()
```

In [24]:
```
model.fit(X_train, Y_train)
```

In [25]:
```
# accuracy score on the training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

In [26]:
```
print('Accuracy score of the training data : ', training_data_accuracy)
```

In [27]:
```
# accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

In [28]:
```
print('Accuracy score of the test data : ', test_data_accuracy)
```

In [29]:
```
X_new = X_test[3]

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

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

In [30]:
```
print(Y_test[3])
```

In [31]:
```
import pickle
```

In [32]:
```
pickle.dump(model,open('model1.pkl','wb'))
model=pickle.load(open('model1.pkl','rb'))
```

In [33]:
```
def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = model.predict(new_xv_test)
    return print(pred_LR)
```

In [34]:
```
import tkinter as tk
from tkinter import *
import pickle
from tkinter import messagebox
```

```
win=tk.Tk()
win.title('FAKE NEWS DETECTOR')
win.geometry("500x300")

win.configure(background="light blue")
scroll=Scrollbar(win)
scroll.pack(side=RIGHT,fill=Y)
img = ImageTk.PhotoImage(Image.open("img6.jpg"))

l=Label(win,text="Fake News Detector",width=200)
l.pack()
canvas = Canvas(
    win,
    width = 500,
    height = 300
    )
canvas.pack(fill='both', expand = True)

canvas.create_image(
    0,
    0,
    image=img,
    anchor = "nw"
    )
```

```
text = Text(
    win,
    height=10,
    bg="light yellow",
    wrap='word',

    width=53,
    yscrollcommand=scroll.set

)
text.place(x=30, y=50)


scroll.config(command=text.yview)
def wordopt(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",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
```

```
def manual_testing():
    news=text.get(0.0,END)

    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = model.predict(new_xv_test)
    k=pred_LR

    if(k==1):

        messagebox.showinfo("Real or Fake",'This is Fake News')


    elif(k==0):

        messagebox.showinfo("Real or Fake",'This is Real News')
    else:

        messagebox.showinfo('real or fake','none')
bt=tk.Button(win,text="Verify",fg="black",bg="white",padx=50,pady=10,command=manual_testing)
bt.place(x=160,y=220)
win.mainloop()
```

Future developments

1.For further development,we can use web scraping and get the data from various social media and websites by ourself and then use them in our system.

2.We can also try to improve the accuracy by query optimisation.

References

- https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model
- https://www.javatpoint.com/data-preprocessing-machine-learning
  Refer Kaggle for train and test data set
- https://www.kaggle.com/c/fake-news/data?select=train.csv