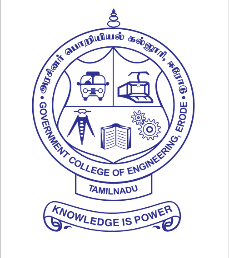
** GOVERNMENT COLLEGE OF ENGINEERING, -ERODE**

**NAAN MUDHAVAN IBM – PROJECT**

**FAKE NEWS DEDECTION USING NLP**

**PROJECT**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**MENTOR**

**Dr.M.POONGOTHAI**

**IT DEPARTMENT**

**TEAM MEMBERS**

**NAME NM-ID**

**1)THANUJ KUMAR.R au731121106050**

**2)SARULATHA.M au731121106041**

**3)KARTHIKA.S au731121106022**

**4)PREETHIGA.A au731121106037**

**5)DIVYADHARSHINI.S au731121106012**

**FAKE NEWS DETECTION USING NLP**

**Importing the libraries**

import pandas as pd

import numpy as np

from sklearn.utils Import shuffle

**Now read the data**

real=pd.read\_csv(‘./Data\_NLP//True.csv.zip’)

fake=pd.read\_csv(‘./Data\_NLP/Fake.csv.zip’)

real.head()

fake.head()

**Now shuffle the data**

real[‘label’]=1

fake[‘label’]=0

news\_data=pdconcat([real,fake],ignore\_index=True)

news\_data=shuffle(news\_data)

news\_data.head()

**Now check the null values in the data set**

news\_data.isna().sum()

**Drop the null values**

news\_data.drop{[‘subject’ , ’date’], axis=1)

**Now clean the stopwords from the dataset**

import re

news\_data[‘text\_processed’]=news\_data[‘text’].map(lambda x: re.sub’(Reuters)’,” ”,x))

news\_data[‘text\_processed’]=news\_data[‘text\_processed’].map(lambda x: re.sub(‘[^A-Za-Z0-9]+’, ’ ‘,x))

news\_data[‘text\_processed’]=news\_data[‘text\_processed’].map(lambda x: x.lower())

news\_data[‘text\_processed’]

**Now import the NLTK and download the wordnet**

import nltk

nltk.download(‘wordnet’)

from nltk.stem import wordNetLemmatizer

def lemmatize(text):

lm=wordNetLemmatizer()

tokens=[lm.lemmatize(word) for word in text.split()]

return “ “.join(tokens)

**Now lemmatize the words**

lemmatize(news\_data[‘text\_processed’] [0])

**Now apply the lemmatize**

news\_data[‘text\_processed’]=news\_data[‘text\_processed’].apply(lemmatize)

from sklearn.feature\_extraction.text import TfidfVectorizer

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

y=news\_data[‘label’]

X=news\_data.drop(‘label’, axis=1)

X.head()

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,stratify=y,random\_state=100)

tfidf=TfidfVectorizer(stop\_words=’english’,ngram\_range=(1,3), lowercase=True, max\_features=5000)

X\_train\_transformed=tfidf.fit\_transform(X\_train[‘text\_processed’])

X\_test\_transformed=tfidf.transform(X\_test[‘text\_processed’])

X\_train\_transformed.shape

**Logistic Regression**

from sklearn.linear\_model import LogisticRegression

lr=LogisticRegression()

lr.fit(X\_train\_transfomed,y\_train)

y\_pred=lr.predict(X\_test\_transformed)

from sklearn.metrics import accuracy\_score,confusion\_matrix

accuracy\_score(y\_test,y\_pred)

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

