Fake news detection using NLP

D.Meghana[1], Anju Haliya[2], B. Sreshta[3]

[1]Department of Computer Science and Engineering, SR University, Warangal, Telangana, India.

[2]Department of Computer Science and Engineering, SR University, Warangal, Telangana, India.

[3]Department of Electronic Communication and Engineering, SR University, Warangal, Telangana, India.

ABSTRACT

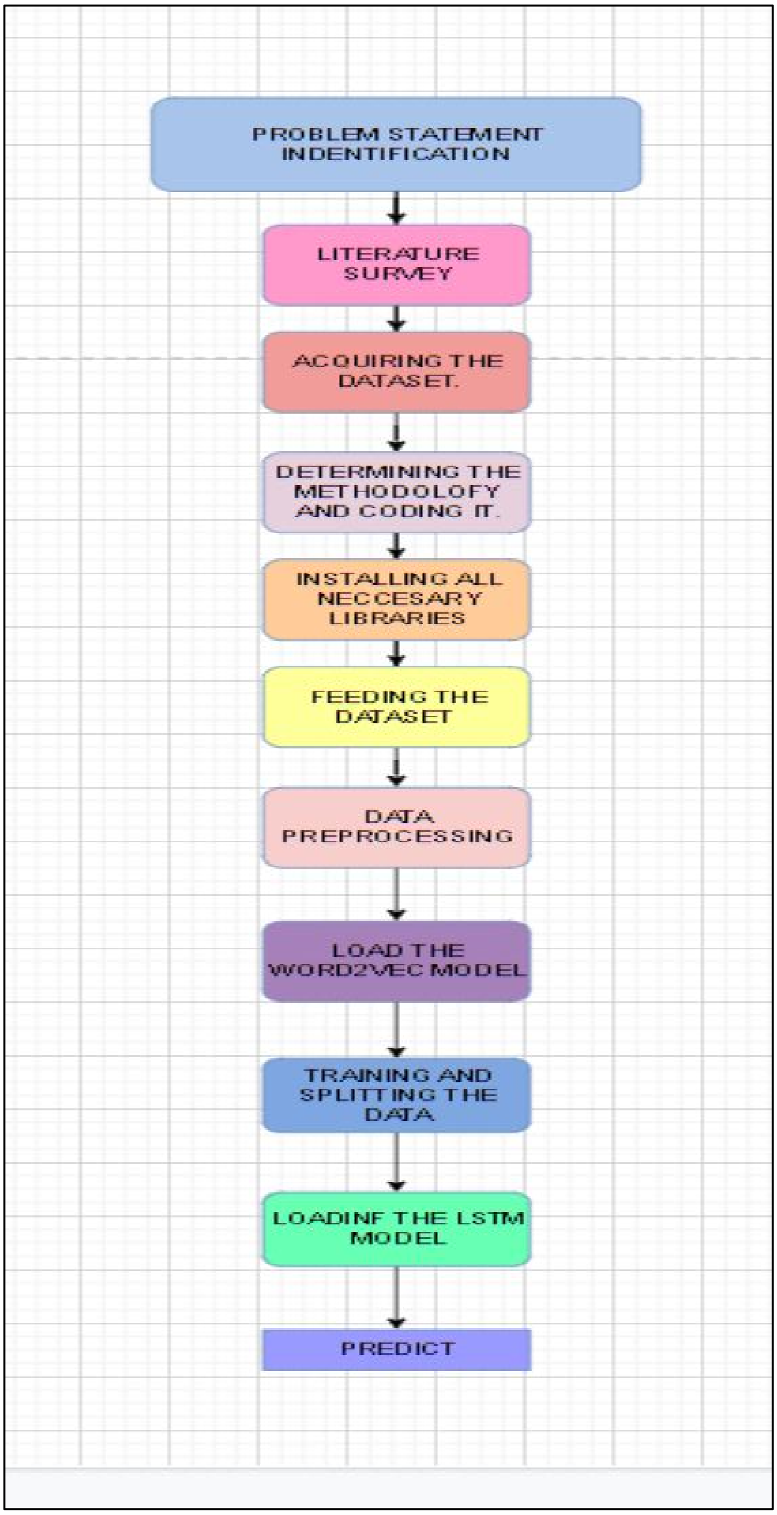
The idea of access to authentic news is the foundation of our project. As internet as evolved to be a dominant reliance for everything and social media has become to be the news providing factor for many. There is rigorous spread of unauthentic or fake news to these chances which are inhibiting the threat to the democracy. The fake news spreading is taken up for much wrong intentions of defamation, back stabbing the legit, biased support for a candidate in election etc. These circumstance would result in threat for an innocent and lack of essential awareness and proper decision making in audience. Moreover, these spread news are made to encash through advertisements by entitling attention grabbing headlines to the click-baits. Therefore, we have understood the necessity to restore the public trust and access to real and genuine news to them. We have referred concepts of NLP - Natural Language Processing for the same and extracted knowledge through our literature survey. We have interfaced the concepts of word2vec and LSTM to carryout word embedding and text classification. To sum up, our model works at a good accuracy of 95 and helps aid the readers with the comfort of the authentic trustworthy news to the best possible.

1. INTRODUCTION

The main goal of our proposed system is to identify the fake news and ensure the benefits of the authentic news to sector of audience because they have a cumulative impact on several aspects of life, our regulations and democracy. Internet has made world a small place to connect collaborate, know things and learn ideologies thereby depending on the same for the news too. Nevertheless, there is a credible authenticity we should be grateful for this technology to smarten our lives in providing news with ease of access.But this has been taken up as an advantage to misuse the resource by spammers, cyber experts, influencers and wicked persons. Alarmingly, the concept of robots being employed to fabricate news and spread lies should be a topic of concern. All these activities are taken forward for various discriminative benefits of an unapplaudable political agenda, diversified notions, monetizing through clicks, saving the accused etc. Therefore, we have realized the necessity to distinguish the fake news from the real news and advance our research accordingly. Then we understood the word embedding concept and used word2vec model for the same. The embedded text is classified using the neural structure of LSTM - Long short term memory because this model is the best for sequential classification.

2. PROBLEM DEFINITION

- Threat to democracy due to fake news.
- Broken public trust due to fake news.
- Achieve reliable justice by real news.
- Understanding the intuition behind the machine learning algorithms of Natural language processing tool kit and its packages.

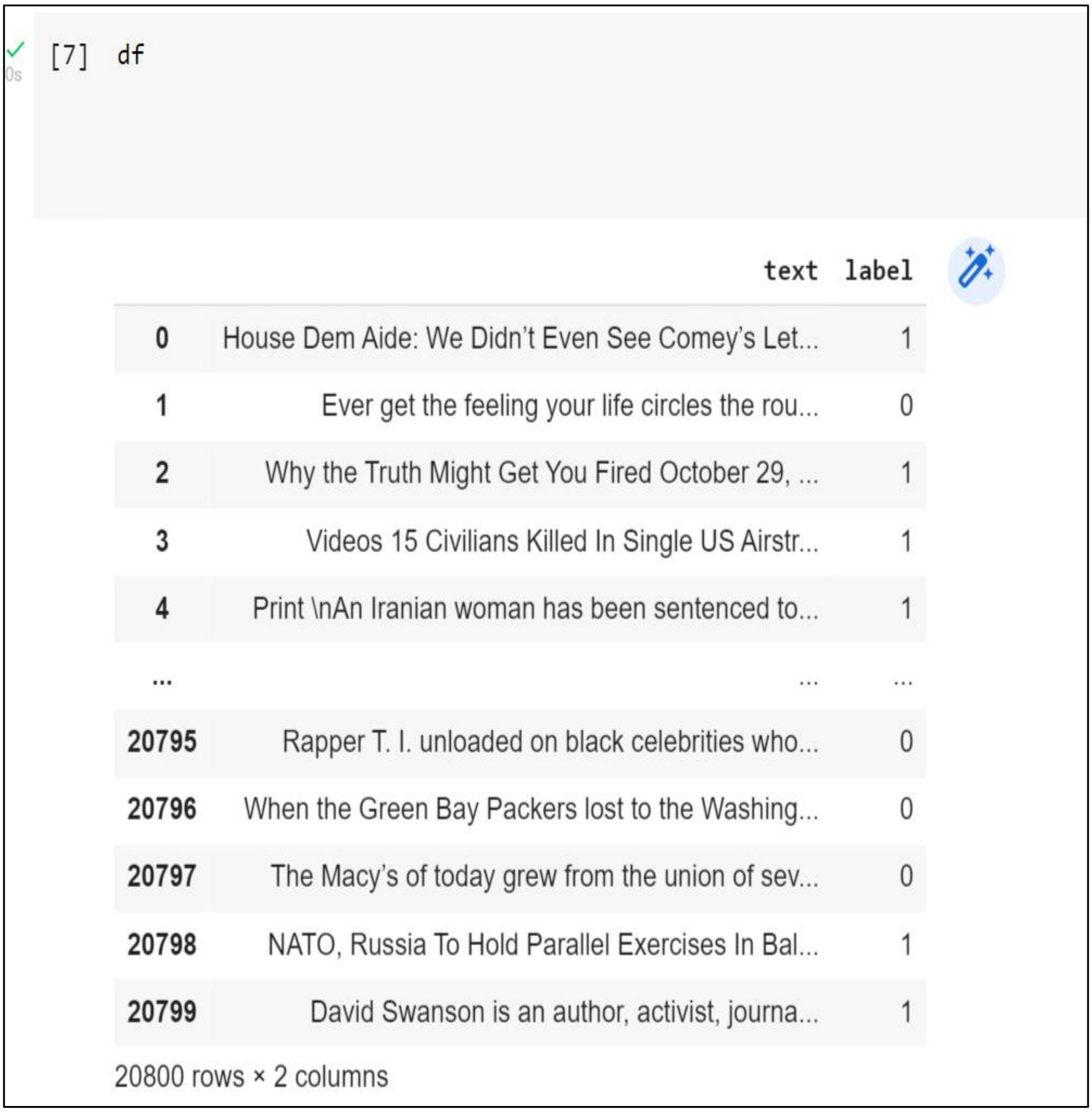

PROBLEM STATEMENT INDENTIFICATION
LITERATURE SURVEY
ACQUIRING THE DATASET.
DETERMINING THE METHODOLOFY AND CODING IT.
INSTALLING ALL NECCESARY LIBRARIES
FEEDING THE DATASET
DATA PREPROCESSING
LOAD THE WORD2VEC MODEL
TRAINING AND SPLITTING THE DATA
LOADINF THE LSTM MODEL
PREDICT

3. DATASET AND ATTRIBUTES

Dataset has been acquired from kaggle :

The thus collected news has features of :
Input features :
>id
> title
>author
>text
>Label (0 for ture 1 for fake)
Output feature :
> real or fake

dataset from kaggle of 20799 samples of fake and true text news



```
[7] df
```

	text	label
0	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	Ever get the feeling your life circles the rou...	0
2	Why the Truth Might Get You Fired October 29, ...	1
3	Videos 15 Civilians Killed In Single US Airstr...	1
4	Print \nAn Iranian woman has been sentenced to...	1
...	...	...
20795	Rapper T. I. unloaded on black celebrities who...	0
20796	When the Green Bay Packers lost to the Washing...	0
20797	The Macy’s of today grew from the union of sev...	0
20798	NATO, Russia To Hold Parallel Exercises In Bal...	1
20799	David Swanson is an author, activist, journa...	1

20800 rows × 2 columns

4. DATA PRE PREOCESSING

Data pre-processing is essential while working on large dataset because algorithms could only be applied on the vectorized text. Data pre - processing thereby aims at covertin text into vectorized simple form which means tokenizing. Tokenizing means dividing the text into units of words or sentences. Tokenizing is the fundamental step for stemming and lemmatizaion.

We have elimated stop wards from the dataset as they have no significance in deciding the meaning of the text. Stemming has been applied to correlate the words belonging to same root.

Then we would proceed with label encoding, the label encoding is to signify the categorical data for the semi-structured or unstructured data. The label encoding means giving the labels for the data in numericals.

Next, stemming is done to produce morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. These algorithms are
used to give the domain vocabularies in domain analysis.

Neural networks requires to have inputs with the same size. Therefore sentence inputs are padded with 0's after defining the max length and words are dropped and added accordingly.

5. METHADOLOGY

Models

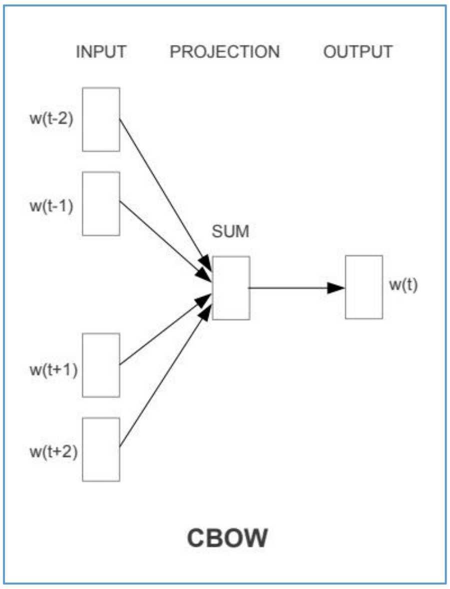
After Data pre-processing we are going to perform word embedding using the Word2Vec vectorizer.

Word2Vec

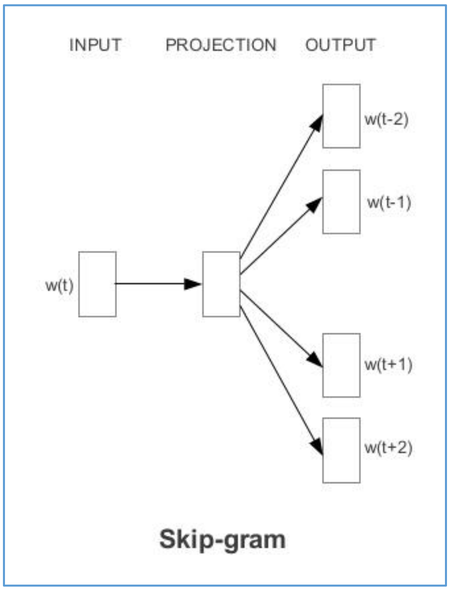
Word2vec is not a singular algorithm, rather, it is a family of model architectures and optimizations that can be used to learn word embeddings from large datasets. Embeddings learned through word2vec have proven to be successful on a variety of downstream natural language processing tasks.

These papers proposed two methods for learning representations of words:

- **Continuous bag-of-words model**: predicts the middle word based on surrounding context words. The context consists of a few words before and after the current (middle) word. This architecture is called a bag-of-words model as the order of words in the context is not important.



- **Continuous skip-gram model**: predicts words within a certain range before and after the current word. A worked example of this is given below.



After word embedding, we are going to load the LSTM model for text classification.

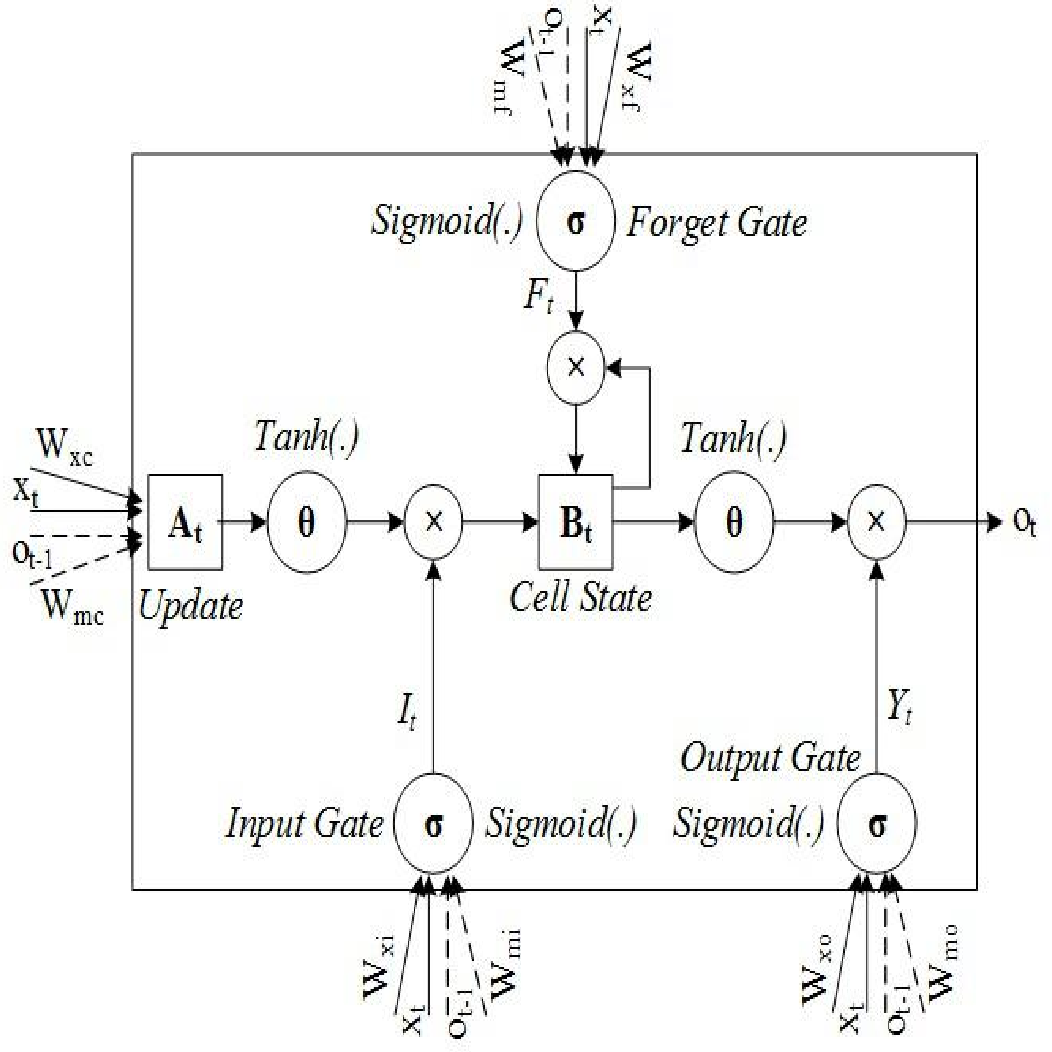
LSTM

LSTM (Long Short-Term Memory) network is a type of RNN (Recurrent Neural Network) that is widely used for learning sequential data prediction problems. As every other neural network LSTM also has some layers which help it to learn and recognize the pattern for better performance. The basic operation of LSTM can be considered to hold the required information and discard the information which is not required or useful for further prediction.

The Architecture of LSTM

A simple LSTM network consists of the following components.

- Forget gate
- Input gate.
- Output gate

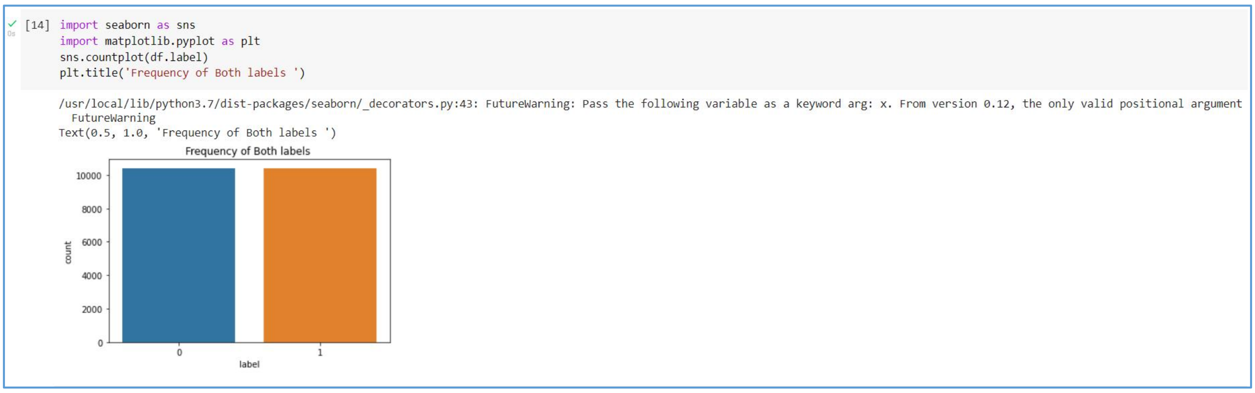


6. RESULTS

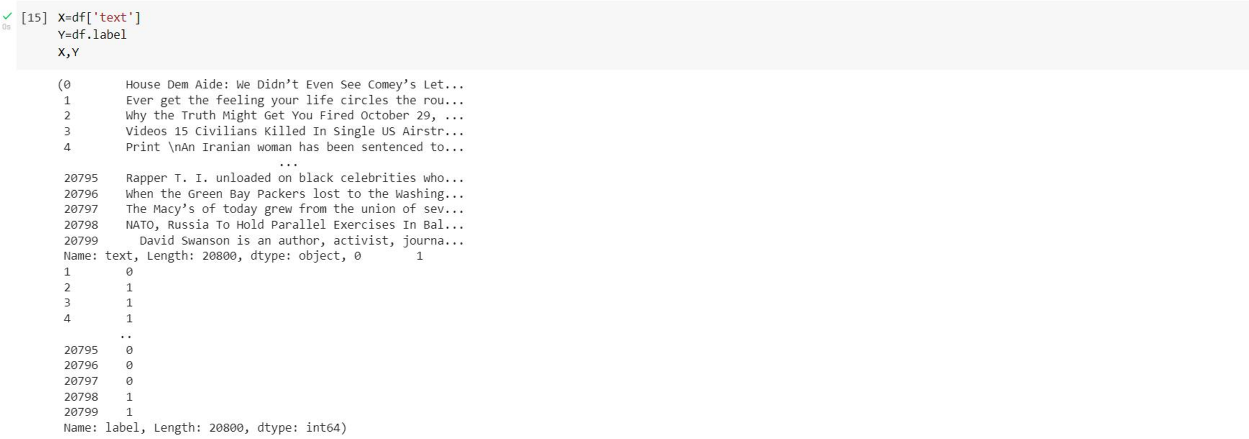
Our project gave out the accuracy of 95%

The output labels are the predictions if the data sample is real or not.

- The dataset containing samples of real and fake news.

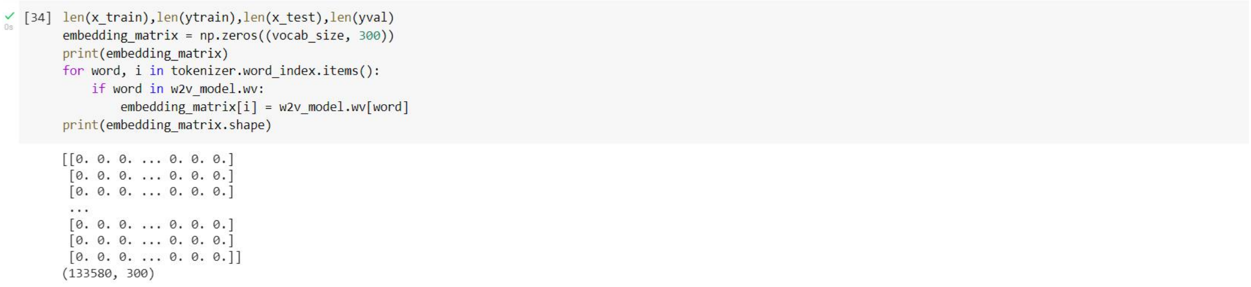


```
[14] import seaborn as sns
     import matplotlib.pyplot as plt
     sns.countplot(df.label)
     plt.title('Frequency of Both labels ')

     /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument
       FutureWarning
     Text(0.5, 1.0, 'Frequency of Both labels ')
```



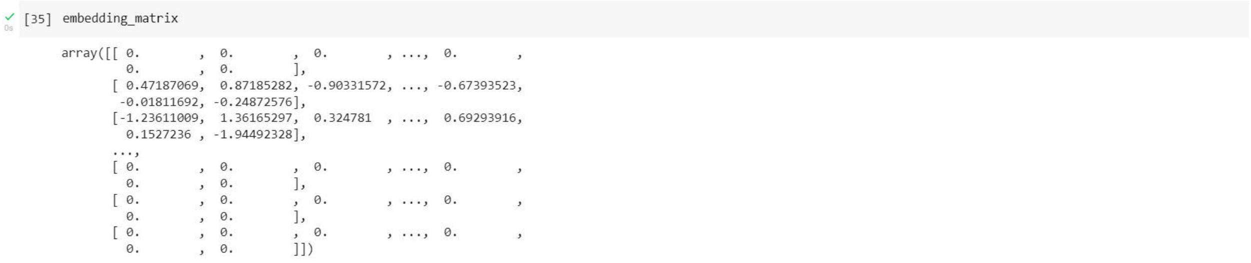
```
[15] X=df['text']
     Y=df.label
     X,Y

     (0        House Dem Aide: We Didn’t Even See Comey’s Let...
      1        Ever get the feeling your life circles the rou...
      2        Why the Truth Might Get You Fired October 29, ...
      3        Videos 15 Civilians Killed In Single US Airstr...
      4        Print \nAn Iranian woman has been sentenced to...
                                     ...
      20795    Rapper T. I. unloaded on black celebrities who...
      20796    When the Green Bay Packers lost to the Washing...
      20797    The Macy’s of today grew from the union of sev...
      20798    NATO, Russia To Hold Parallel Exercises In Bal...
      20799      David Swanson is an author, activist, journa...
      Name: text, Length: 20800, dtype: object, 0        1
      1        0
      2        1
      3        1
      4        1
              ..
      20795    0
      20796    0
      20797    0
      20798    1
      20799    1
      Name: label, Length: 20800, dtype: int64)
```

- # Word embedding results



```
[34] len(x_train),len(ytrain),len(x_test),len(yval)
     embedding_matrix = np.zeros((vocab_size, 300))
     print(embedding_matrix)
     for word, i in tokenizer.word_index.items():
         if word in w2v_model.wv:
             embedding_matrix[i] = w2v_model.wv[word]
     print(embedding_matrix.shape)
```

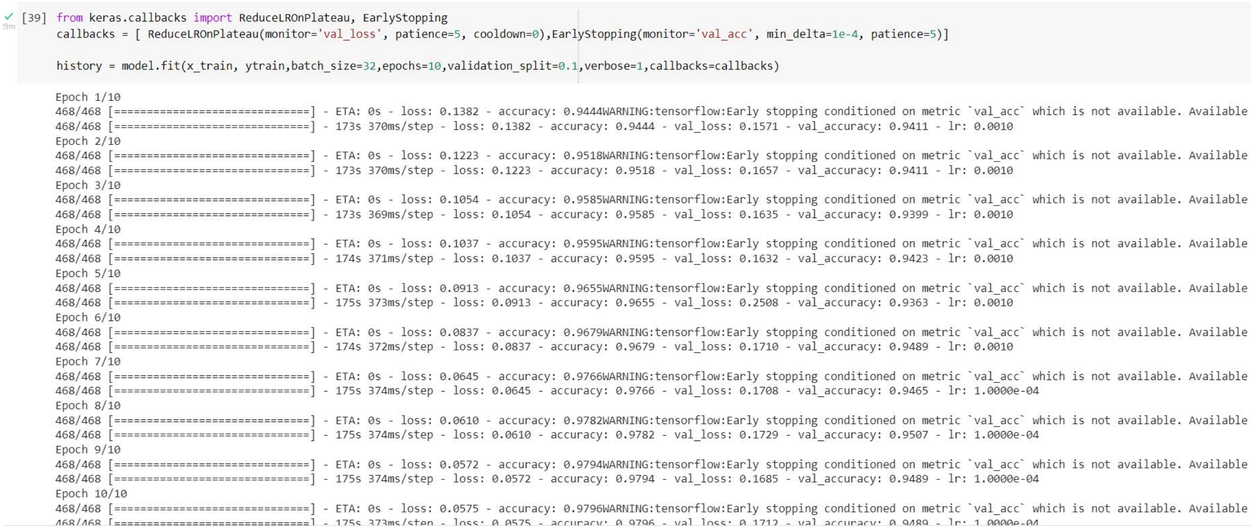
```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
(133580, 300)
```



```
[35] embedding_matrix
```

```
array([[ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.47187069,  0.87185282, -0.90331572, ..., -0.67393523,
        -0.01811692, -0.24872576],
       [-1.23611009,  1.36165297,  0.324781  , ...,  0.69293916,
         0.1527236 , -1.94492328],
       ...,
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ]])
```

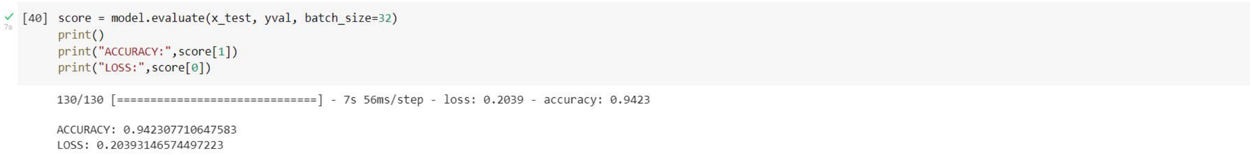
- # LSTM model results



```
[39] from keras.callbacks import ReduceLROnPlateau, EarlyStopping
     callbacks = [ ReduceLROnPlateau(monitor='val_loss', patience=5, cooldown=0),EarlyStopping(monitor='val_acc', min_delta=1e-4, patience=5)]

     history = model.fit(x_train, ytrain,batch_size=32,epochs=10,validation_split=0.1,verbose=1,callbacks=callbacks)
```

```
Epoch 1/10
468/468 [==============================] - ETA: 0s - loss: 0.1382 - accuracy: 0.9444WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 173s 370ms/step - loss: 0.1382 - accuracy: 0.9444 - val_loss: 0.1571 - val_accuracy: 0.9411 - lr: 0.0010
Epoch 2/10
468/468 [==============================] - ETA: 0s - loss: 0.1223 - accuracy: 0.9518WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 173s 370ms/step - loss: 0.1223 - accuracy: 0.9518 - val_loss: 0.1657 - val_accuracy: 0.9411 - lr: 0.0010
Epoch 3/10
468/468 [==============================] - ETA: 0s - loss: 0.1054 - accuracy: 0.9585WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 173s 369ms/step - loss: 0.1054 - accuracy: 0.9585 - val_loss: 0.1635 - val_accuracy: 0.9399 - lr: 0.0010
Epoch 4/10
468/468 [==============================] - ETA: 0s - loss: 0.1037 - accuracy: 0.9595WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 174s 371ms/step - loss: 0.1037 - accuracy: 0.9595 - val_loss: 0.1632 - val_accuracy: 0.9423 - lr: 0.0010
Epoch 5/10
468/468 [==============================] - ETA: 0s - loss: 0.0913 - accuracy: 0.9655WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 175s 373ms/step - loss: 0.0913 - accuracy: 0.9655 - val_loss: 0.2508 - val_accuracy: 0.9363 - lr: 0.0010
Epoch 6/10
468/468 [==============================] - ETA: 0s - loss: 0.0837 - accuracy: 0.9679WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 174s 372ms/step - loss: 0.0837 - accuracy: 0.9679 - val_loss: 0.1710 - val_accuracy: 0.9489 - lr: 0.0010
Epoch 7/10
468/468 [==============================] - ETA: 0s - loss: 0.0645 - accuracy: 0.9766WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 175s 374ms/step - loss: 0.0645 - accuracy: 0.9766 - val_loss: 0.1708 - val_accuracy: 0.9465 - lr: 1.0000e-04
Epoch 8/10
468/468 [==============================] - ETA: 0s - loss: 0.0610 - accuracy: 0.9782WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 175s 374ms/step - loss: 0.0610 - accuracy: 0.9782 - val_loss: 0.1729 - val_accuracy: 0.9507 - lr: 1.0000e-04
Epoch 9/10
468/468 [==============================] - ETA: 0s - loss: 0.0572 - accuracy: 0.9794WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 175s 374ms/step - loss: 0.0572 - accuracy: 0.9794 - val_loss: 0.1685 - val_accuracy: 0.9489 - lr: 1.0000e-04
Epoch 10/10
468/468 [==============================] - ETA: 0s - loss: 0.0575 - accuracy: 0.9796WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available
468/468 [==============================] - 175s 373ms/step - loss: 0.0575 - accuracy: 0.9796 - val_loss: 0.1712 - val_accuracy: 0.9489 - lr: 1.0000e-04
```

- # ACCURACY



```
[40] score = model.evaluate(x_test, yval, batch_size=32)
     print()
     print("ACCURACY:",score[1])
     print("LOSS:",score[0])
```

```
130/130 [==============================] - 7s 56ms/step - loss: 0.2039 - accuracy: 0.9423

ACCURACY: 0.942307710647583
LOSS: 0.20393146574497223
```

7. CONCLUSION

To conclude, our project meets its objective of counter the threat to deomocracty through the fake news and restore the public trust and give them access to real news for proper awareness and decision making. Our model uses the tensorflow and keras libraries for data preprocessing, text, embedding and classification. The data preprocessing has been done by tokenizing, label encoding, stemming, pad sequencing methods and word embedding is done by word2vec model from Gensim and text classification between true and fake by LSTM - long short term memory libabry. Our model accounts for an accuracy of 95%.

8. REFERENCES

[1] https://www.ijraset.com/research-paper/paper-on-fake-news-detection-using-machine-learning

[2] https://ieeexplore.ieee.org/document/8546944

[3] https://www.sciencedirect.com/science/article/pii/S2667096820300070

[4] https://www.ijraset.com/research-paper/paper-on-fake-news-detection-using-machine-learning

[5] https://www.academia.edu/es/51346745/Fake_News_Detection_using_LSTM

[6] https://www.analyticsvidhya.com/blog/2021/07/detecting-fake-news-with-natural-language-processing

[7] https://towardsdatascience.com/predicting-fake-news-using-nlp-and-machine-learning-scikit-learn-glove-keras-lstm-7bbd557c3443?gi=4de371a5a91e

[8] https://www.sciencedirect.com/science/article/pii/S187705092101797X

[9] https://link.springer.com/article/10.1007/s42979-021-00775-6

[10] https://medium.com/analytics-vidhya/fake-news-detection-using-nlp-techniques-c2dc4be05f99