FLIP ROBO

FAKE NEWS PROJECT REPORT

Submitted by:

NEETAL TIWARI

ACKNOWLEDGMENT

I would like to express my special thank of gratitude to my SME (Mohd Kashif) as well as my company (Flip Robo Technologies) who gave me the golden opportunity to do this wonderful project on the (Fake news project) which also helped me to doing lots of research and I came to know about so many things. I am really thankful to them.

INTRODUCTION

Business Problem Framing

Fake News has become one of the major problems in the existing society. Fake News has high potential to change opinions, facts and can be the most dangerous weapon in influencing society. The proposed project uses NLP techniques for detecting the 'fake news', that is, misleading news stories which come from the non-reputable sources. By building a model based on a Decision Tree Classifier algorithm, the fake news can be detected. The data science community has responded by taking actions against the problem. It is impossible to determine a news as real or fake accurately. So, the proposed project uses the datasets that are trained using count vectorizer method for the detection of fake news and its accuracy will be tested using machine learning algorithms.

Conceptual Background of the Domain Problem

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.

Review of Literature

The data set includes:

title: The Title of a news article

text: The text of the title

subject: Subject of the news article

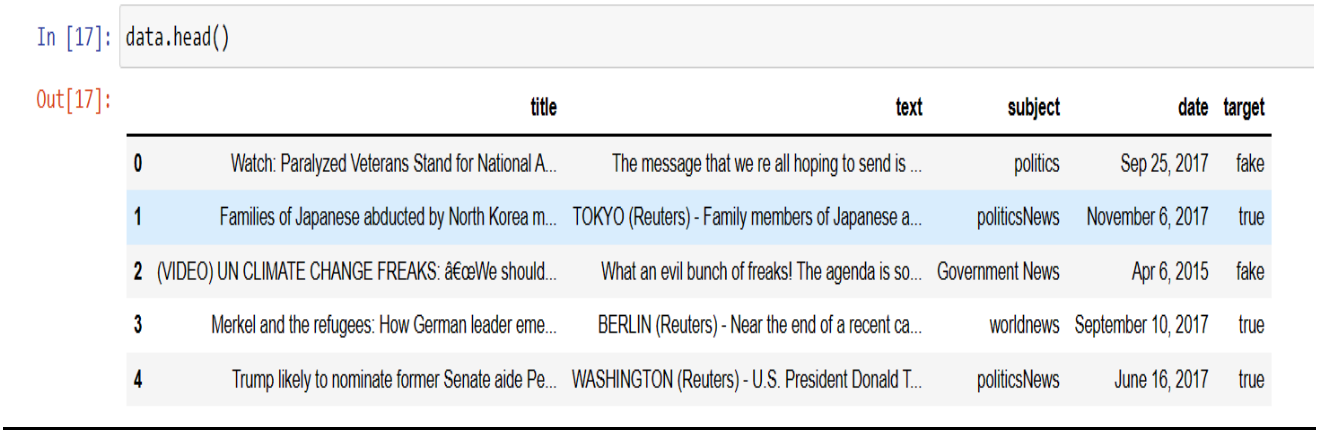
date: Date of the News article

target: A label that marks the article is fake or true

Motivation for the Problem Undertaken

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 1 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 some significant progress towards that goal.

Data Sources and their formats



```
In [17]: data.head()
```

Out[17]:

	title	text	subject	date	target
0	Watch: Paralyzed Veterans Stand for National A...	The message that we re all hoping to send is ...	politics	Sep 25, 2017	fake
1	Families of Japanese abducted by North Korea m...	TOKYO (Reuters) - Family members of Japanese a...	politicsNews	November 6, 2017	true
2	(VIDEO) UN CLIMATE CHANGE FREAKS: â€œWe should...	What an evil bunch of freaks! The agenda is so...	Government News	Apr 6, 2015	fake
3	Merkel and the refugees: How German leader eme...	BERLIN (Reuters) - Near the end of a recent ca...	worldnews	September 10, 2017	true
4	Trump likely to nominate former Senate aide Pe...	WASHINGTON (Reuters) - U.S. President Donald T...	politicsNews	June 16, 2017	true

We can see in fake project dataset 5 columns are there.

The data set includes:

title: The Title of a news article

text: The text of the title

subject: Subject of the news article

date: Date of the News article

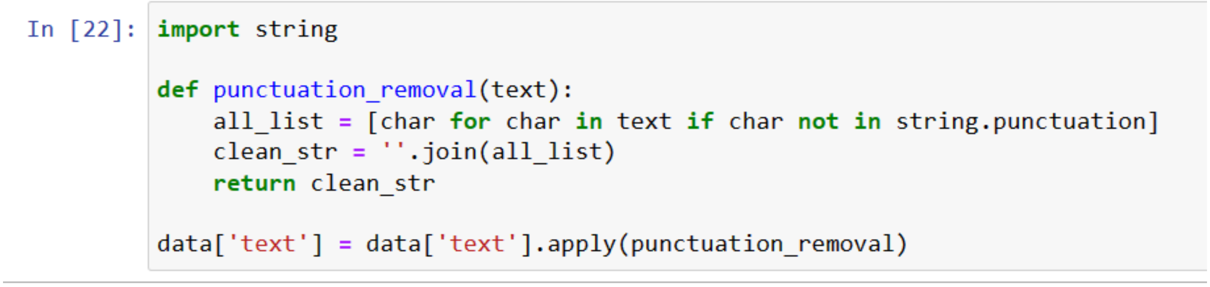
target: A label that marks the article is fake or true

Data Pre-processing Done

In any Machine Learning process, Data Pre-processing is that step in which the data gets transformed, or encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm. In this fake news detection, pre-processing is the major thing that should be done. Firstly, as the data dataset is collected from various sources unnecessary information should be removed, converted to lower case, remove punctuation, symbols, stop words.

Removing punctuation

The punctuation removal process will help to treat each text equally. For example, the word data and data! are treated equally after the process of removal of punctuations.



```
In [22]: import string

         def punctuation_removal(text):
             all_list = [char for char in text if char not in string.punctuation]
             clean_str = ''.join(all_list)
             return clean_str

         data['text'] = data['text'].apply(punctuation_removal)
```

STOP WORD REMOVAL

A Stop Word is a commonly used word in any natural language such as “a, an, the, for, is, was, which, are, were, from, do, with, and, so, very, that, this, no, yourselves etc....”. These Stop Words will have a very high frequency and so these should be eliminated while calculating the term frequency so that the other important things are given priority. Stop word removal is such a Pre-processing step which removes these stop words and thereby helping in the further steps and also reducing some processing time because the size of the document decreases tremendously.

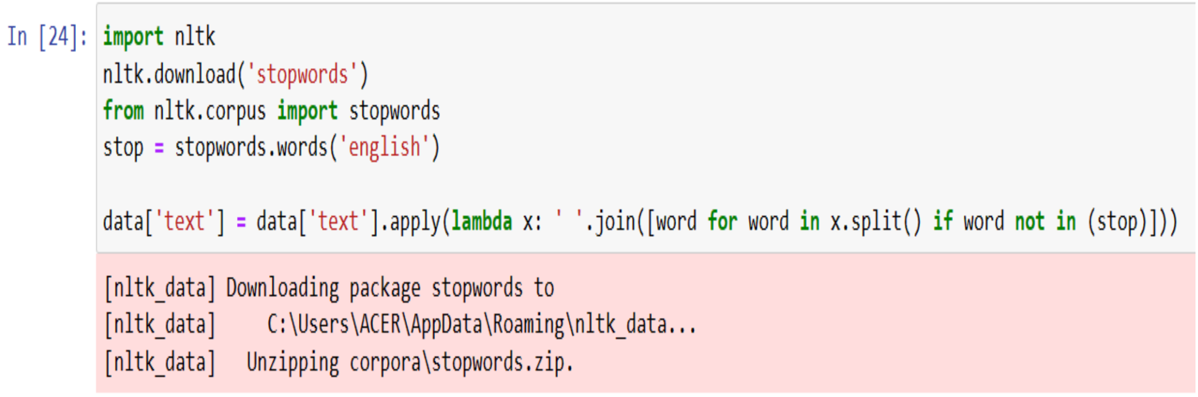
Consider a Sentence

“This is a sample sentence, showing off the stop word removal”.

Output after Stop word removal is:

[“sample”, “sentence”, “showing”, “stop”, “word”, “removal”]

Note: Though Stop words refer to the most commonly used words in a particular language, there is no single universal list of stop words, different tools use different stop words.



```
In [24]: import nltk
         nltk.download('stopwords')
         from nltk.corpus import stopwords
         stop = stopwords.words('english')

         data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
```

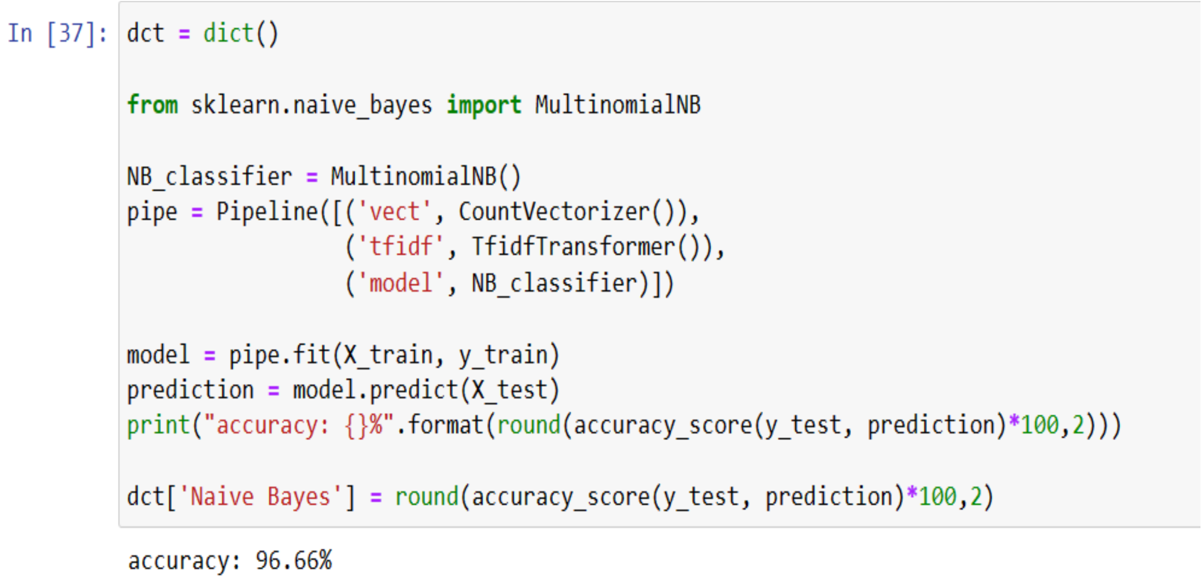
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\ACER\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Model/s Development and Evaluation

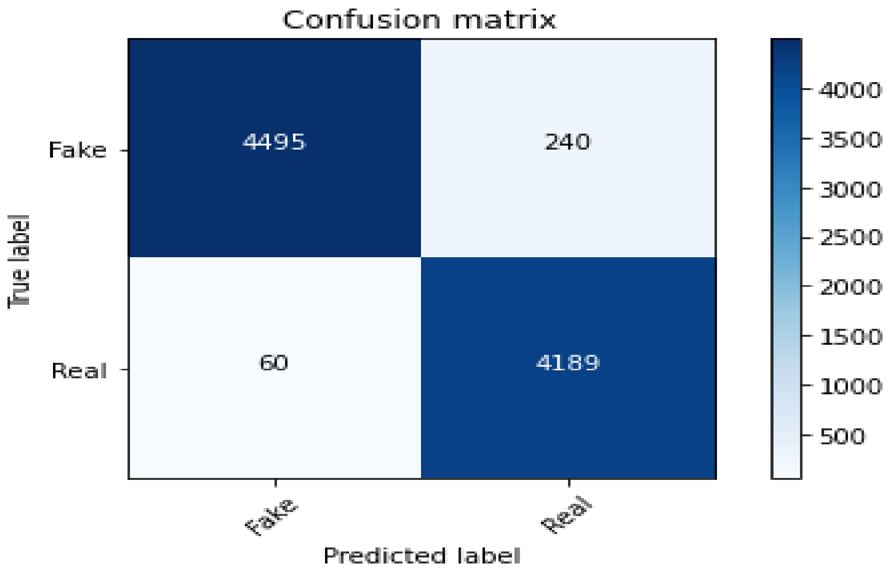
Testing of Identified Approaches (Algorithms)

For data analysis I had used five algorithms such as Naive Bayes, Logistic regression, DecisionTreeClassifier, RandomForestClassifier, SVM

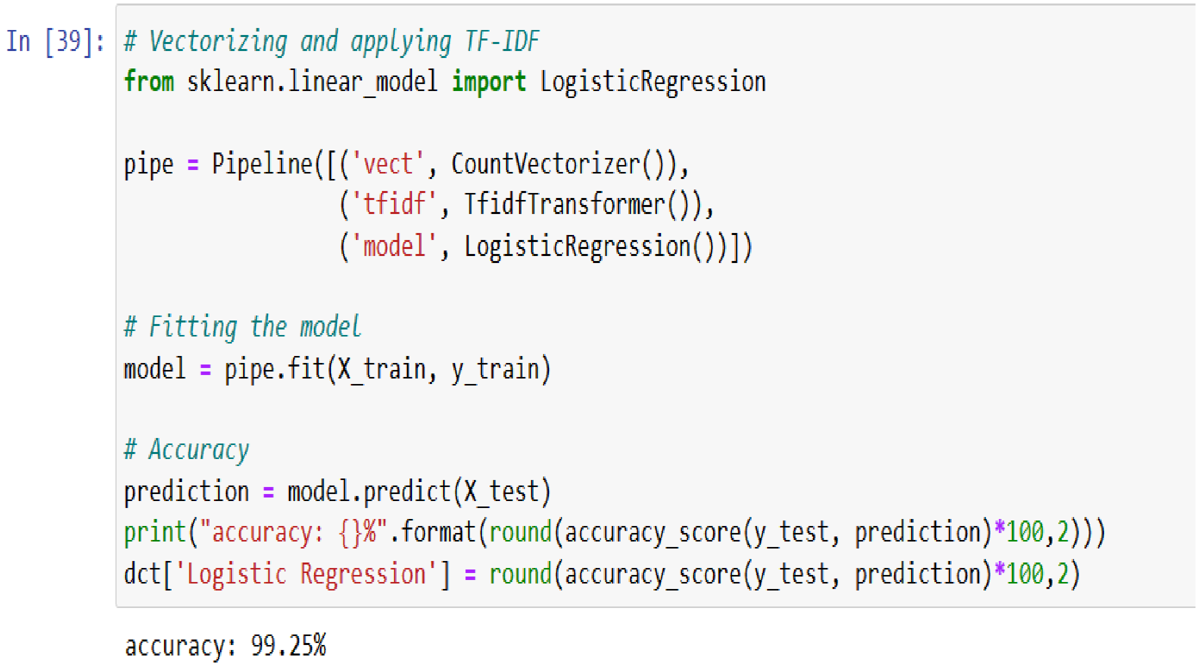
Run and evaluate selected models



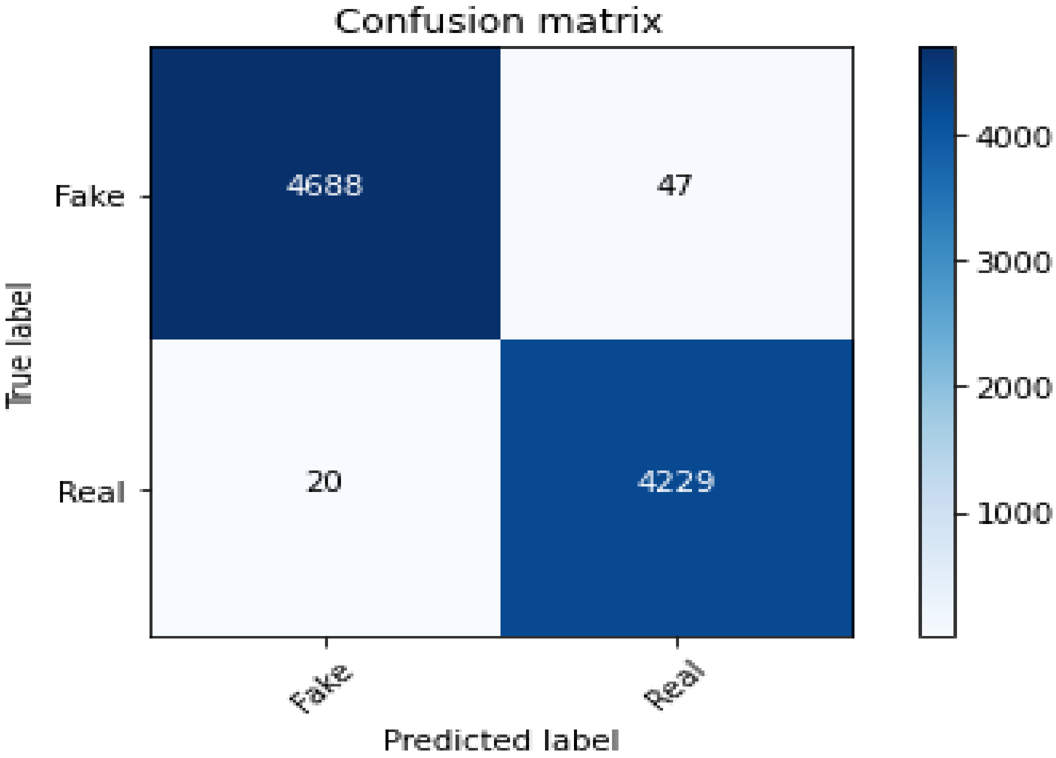
```
In [37]: dct = dict()

         from sklearn.naive_bayes import MultinomialNB

         NB_classifier = MultinomialNB()
         pipe = Pipeline([('vect', CountVectorizer()),
                          ('tfidf', TfidfTransformer()),
                          ('model', NB_classifier)])

         model = pipe.fit(X_train, y_train)
         prediction = model.predict(X_test)
         print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))

         dct['Naive Bayes'] = round(accuracy_score(y_test, prediction)*100,2)

         accuracy: 96.66%
```



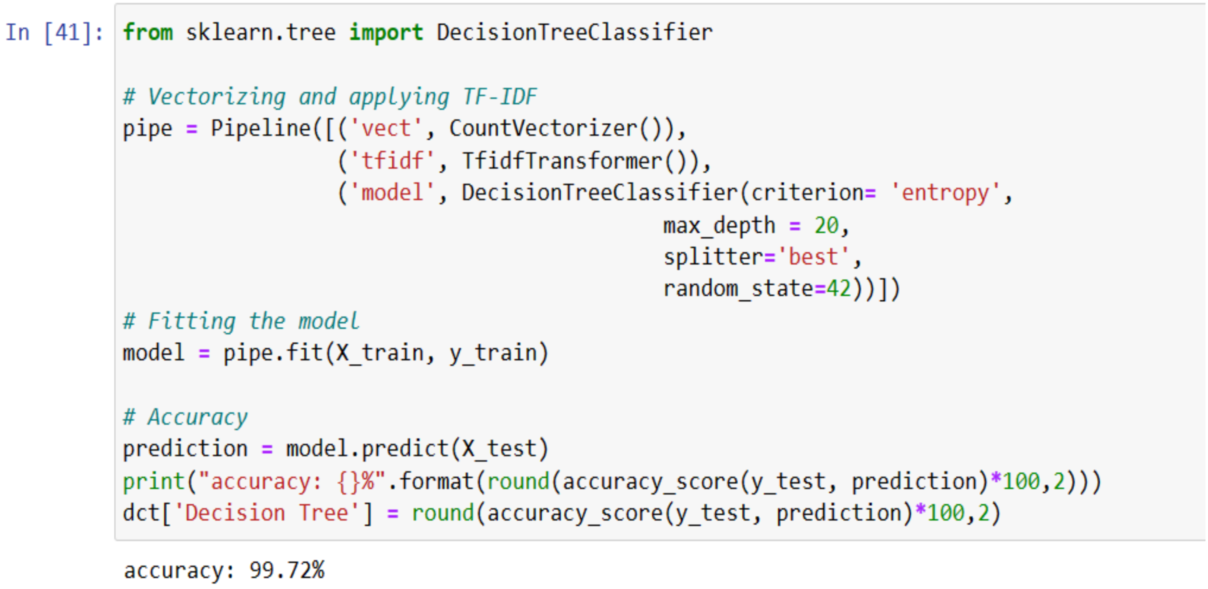
Accuracy score of NAIVE BAYES is 96.66%



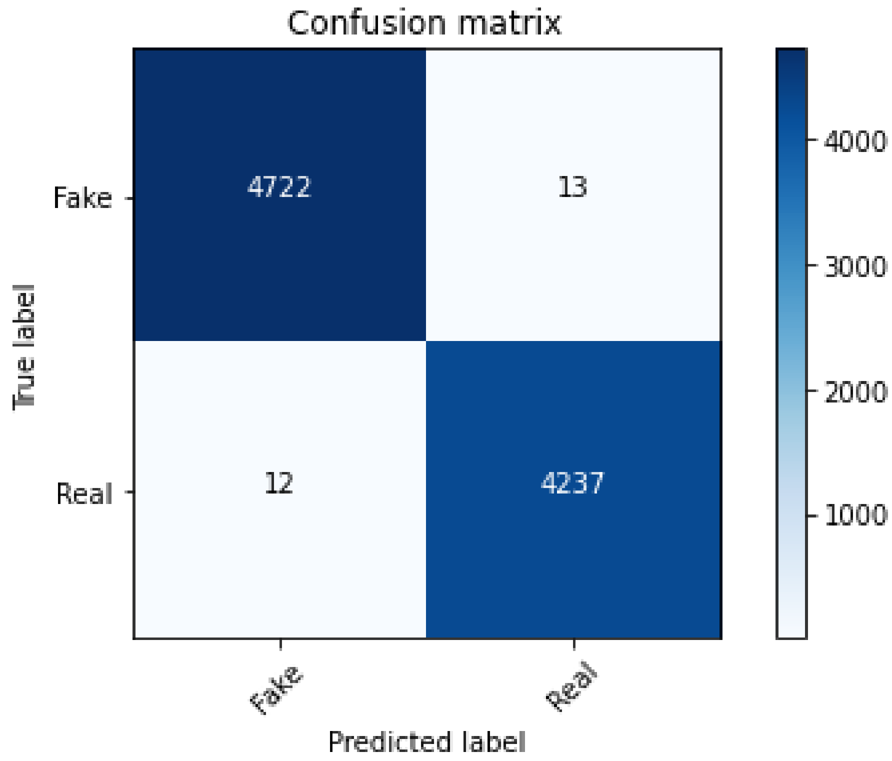
```
In [39]: # Vectorizing and applying TF-IDF
         from sklearn.linear_model import LogisticRegression

         pipe = Pipeline([('vect', CountVectorizer()),
                          ('tfidf', TfidfTransformer()),
                          ('model', LogisticRegression())])

         # Fitting the model
         model = pipe.fit(X_train, y_train)

         # Accuracy
         prediction = model.predict(X_test)
         print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
         dct['Logistic Regression'] = round(accuracy_score(y_test, prediction)*100,2)

         accuracy: 99.25%
```



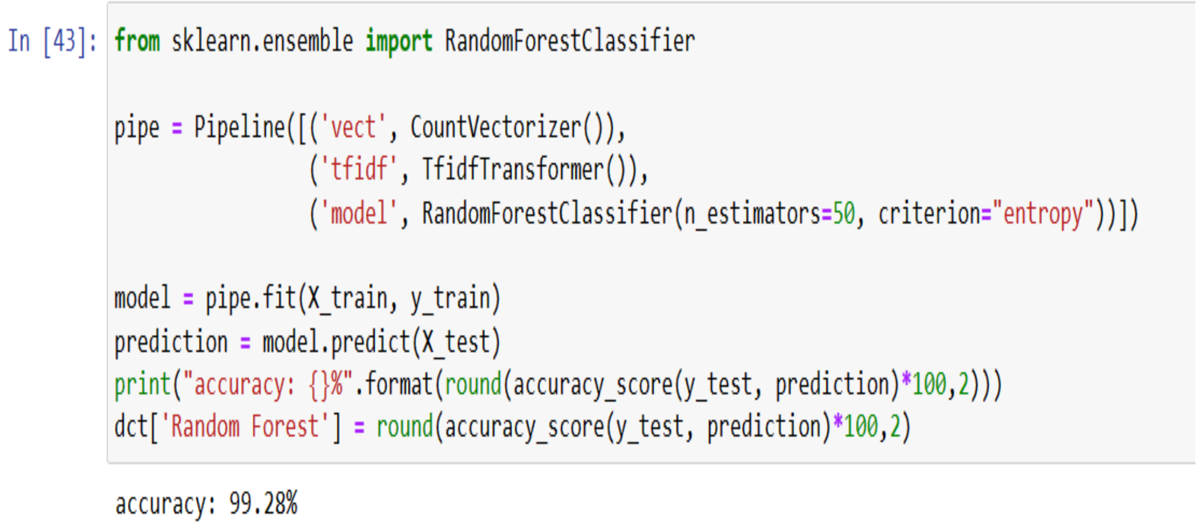
Accuracy score of Logistic regression is 99.25%



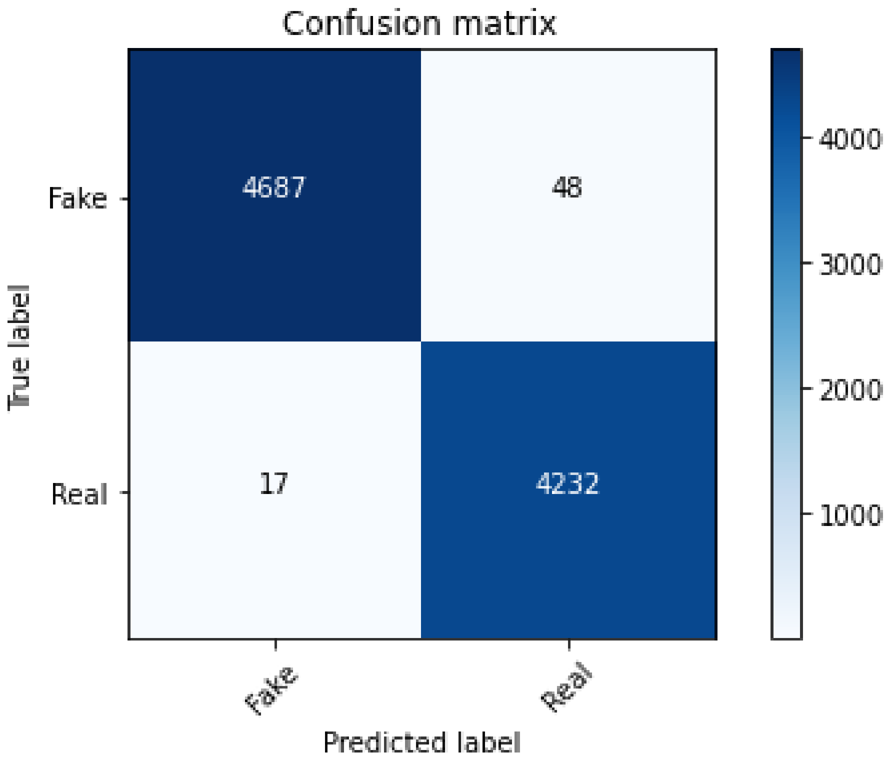
```
In [41]: from sklearn.tree import DecisionTreeClassifier

         # Vectorizing and applying TF-IDF
         pipe = Pipeline([('vect', CountVectorizer()),
                          ('tfidf', TfidfTransformer()),
                          ('model', DecisionTreeClassifier(criterion= 'entropy',
                                                  max_depth = 20,
                                                  splitter='best',
                                                  random_state=42))])
         # Fitting the model
         model = pipe.fit(X_train, y_train)

         # Accuracy
         prediction = model.predict(X_test)
         print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
         dct['Decision Tree'] = round(accuracy_score(y_test, prediction)*100,2)

         accuracy: 99.72%
```



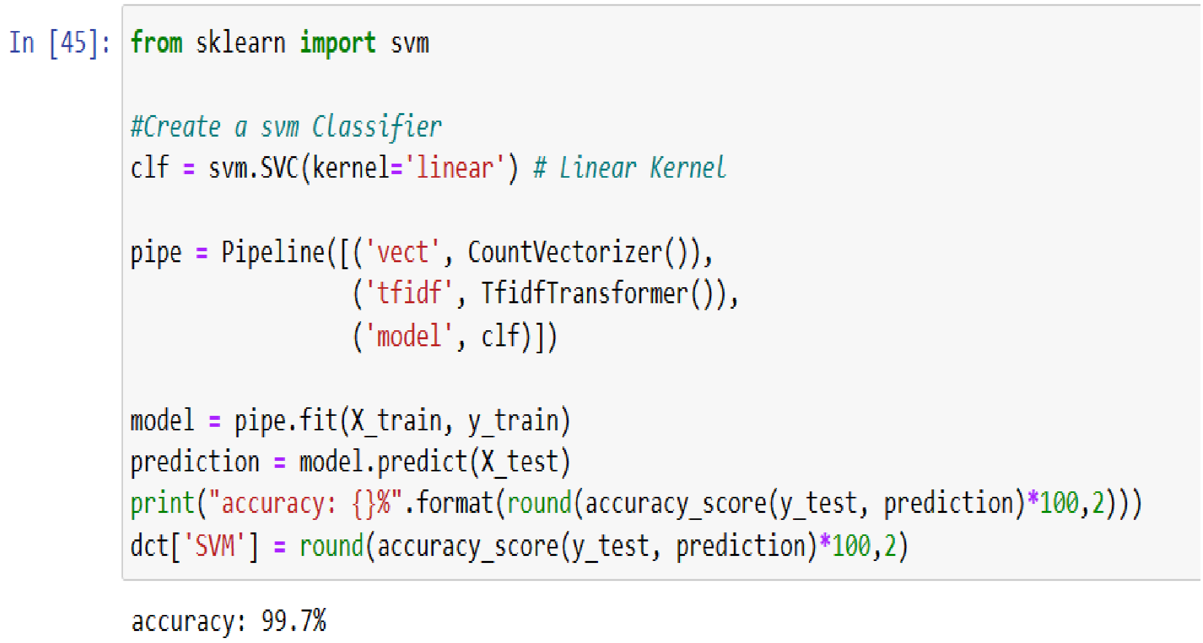
Accuracy score of Decision tree classifier is 99.72%



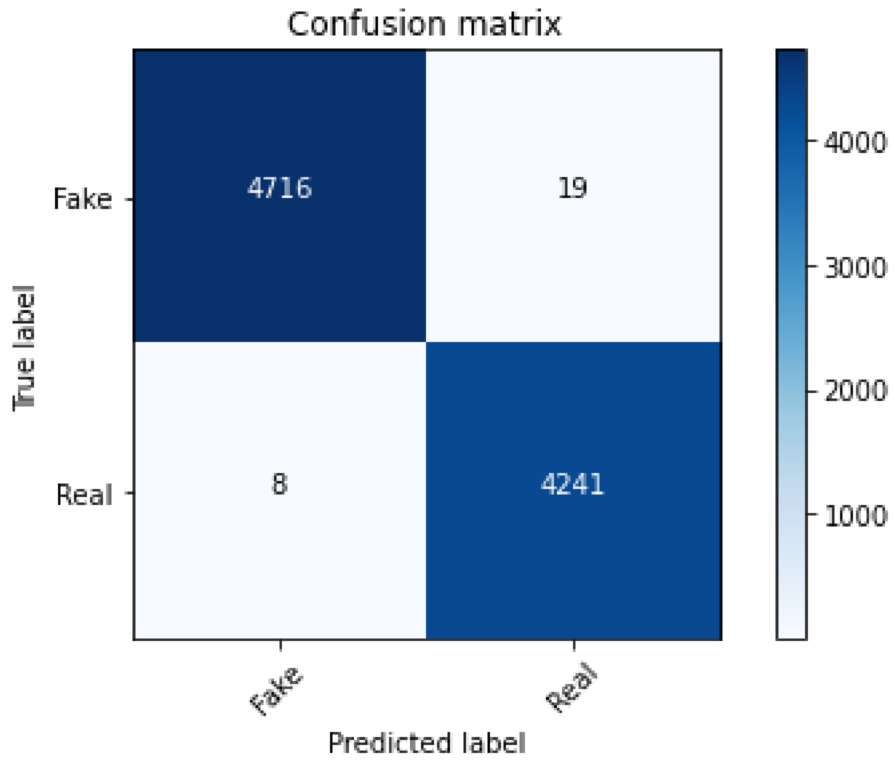
```
In [43]: from sklearn.ensemble import RandomForestClassifier

         pipe = Pipeline([('vect', CountVectorizer()),
                          ('tfidf', TfidfTransformer()),
                          ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])

         model = pipe.fit(X_train, y_train)
         prediction = model.predict(X_test)
         print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
         dct['Random Forest'] = round(accuracy_score(y_test, prediction)*100,2)

         accuracy: 99.28%
```



Accuracy score of Random Forest classifier is 99.28%



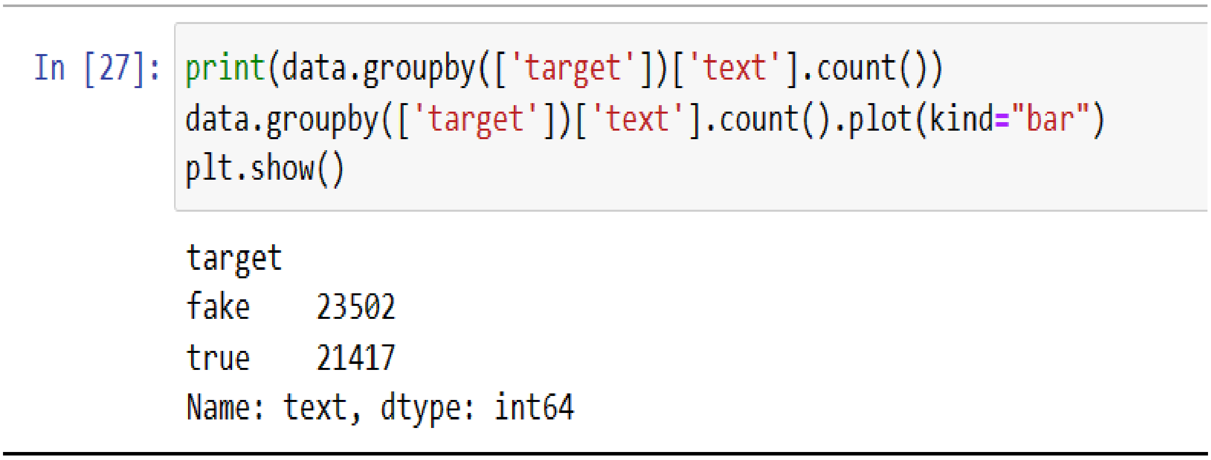
```
In [45]: from sklearn import svm

         #Create a svm Classifier
         clf = svm.SVC(kernel='linear') # Linear Kernel

         pipe = Pipeline([('vect', CountVectorizer()),
                          ('tfidf', TfidfTransformer()),
                          ('model', clf)])

         model = pipe.fit(X_train, y_train)
         prediction = model.predict(X_test)
         print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
         dct['SVM'] = round(accuracy_score(y_test, prediction)*100,2)

         accuracy: 99.7%
```



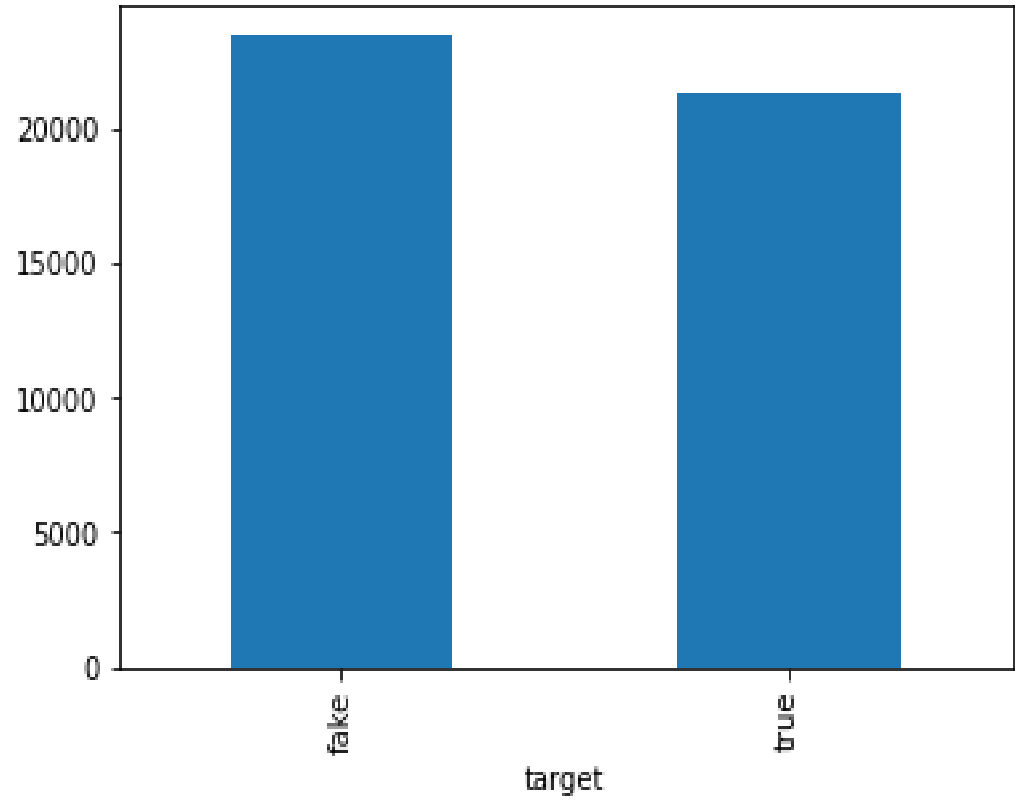
Accuracy Score of SVM is 99.7%

Visualizations

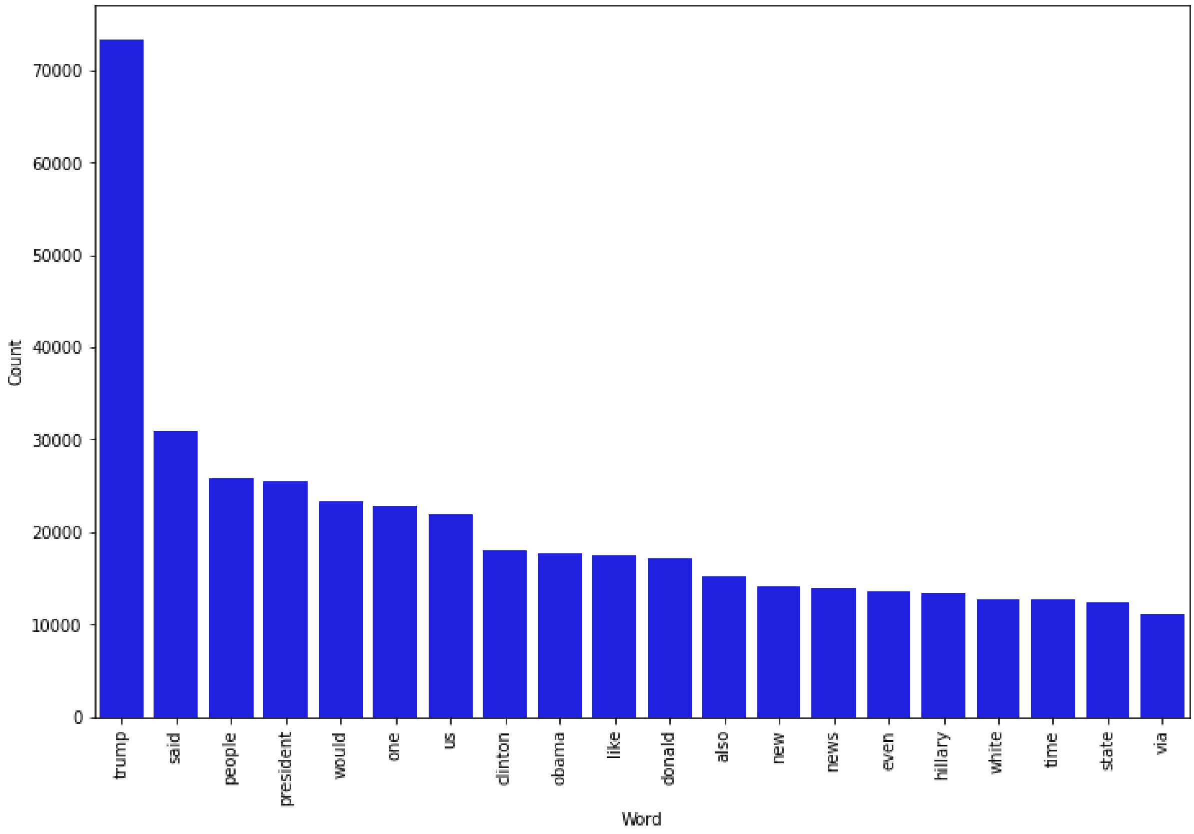
For visualization I had used matplotlib.pyplot and seaborn modules.



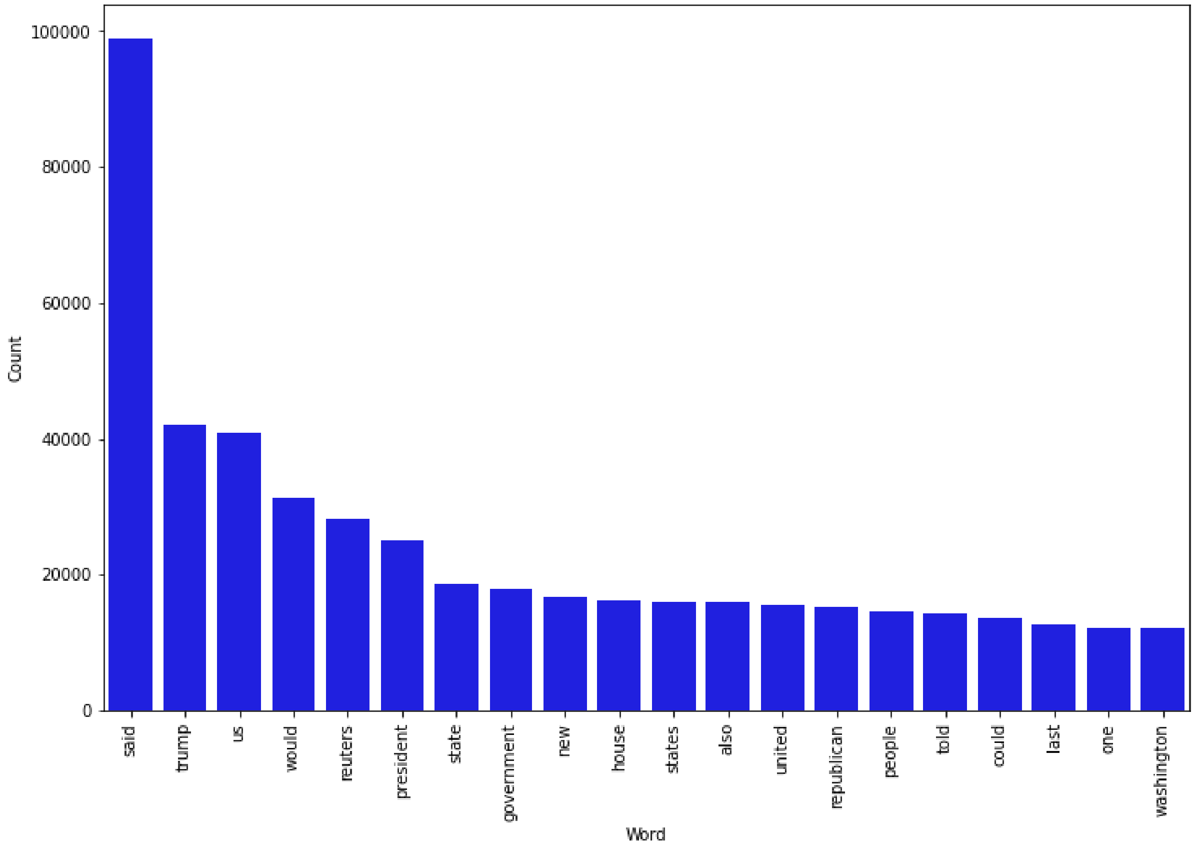
```
In [27]: print(data.groupby(['target'])['text'].count())
         data.groupby(['target'])['text'].count().plot(kind="bar")
         plt.show()

         target
         fake    23502
         true    21417
         Name: text, dtype: int64
```



fake news is 23502 and true news are 21417.



These all are most frequent words in fake news. trump word is highly used in fake news.



These all are most frequent words in true news. said word is highly used in true news.

CONCLUSION

Our project can ring the initial alert for fake news. The model produces worse results if the article is written cleverly, without any denationalization. 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.