MACHINE LEARNING APPLIED: FAKE NEWS CLASSIFICATION

Submitted by:

S. Dilip Kumar

ACKNOWLEDGMENT

Foremost, I would like to express my sincere gratitude to Data Trained team for the continuous support of my Data Science study and research, for the patience, motivation, enthusiasm, and immense knowledge. The guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Data science study.

Besides Data Trained, I would like to thank Flip Robo Team, for their encouragement, insightful internship, and help to understand the study.

My sincere thanks also go to SME Shubham Yadav, Mathur Rishi, and Vaishali Singh, Khushboo Garg for offering me the internship opportunities in their ally and leading me working on diverse exciting projects. I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

Last but not the least, I would like to thank my family, my parents, for being a morale support to me at the first place and backing me up in any down time, for being there in all hard situations making me to fight against any difficulty I face.

INTRODUCTION

Business Problem Framing

Fake news is false or misleading information presented as news. It often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue. However, the term does not have a fixed definition, and has been applied more broadly to include any type of false information, including unintentional and unconscious mechanisms, and also by high-profile individuals to apply to any news unfavourable to his/her personal perspectives.

Once common in print, the prevalence of fake news has increased with the rise of social media, especially the Facebook News Feed. Political polarization, post-truth politics, confirmation bias, and social media algorithms have been implicated in the spread of fake news. It is sometimes generated and propagated by hostile foreign actors, particularly during elections. The use of anonymously-hosted fake news websites has made it difficult to prosecute sources of fake news for libel. In some definitions, fake news includes satirical articles misinterpreted as genuine, and articles that employ sensationalist or clickbait headlines that are not supported in the text.

Fake news can reduce the impact of real news by competing with it; a Buzzfeed analysis found that the top fake news stories about the 2016 U.S. presidential election received more engagement on Facebook than top stories from major media outlets. It also has the potential to undermine trust in serious media coverage. The term has at times been used to cast doubt upon legitimate news, and former U.S. president Donald Trump has been credited with popularizing the term by using it to describe any negative press coverage of himself. It has been increasingly criticized, due in part to Trump's misuse, with the British government deciding to avoid the term, as it is "poorly-

defined" and "conflates a variety of false information, from genuine error through to foreign interference".

Multiple strategies for fighting fake news are currently being actively researched, and need to be tailored to individual types of fake news. Effective self-regulation and legally-enforced regulation of social media and web search engines are needed. The information space needs to be flooded with accurate news to displace fake news. Individuals need to actively confront false narratives when spotted, as well as take care when sharing information via social media. However, reason, the scientific method and critical thinking skills alone are insufficient to counter the broad scope of bad ideas. Overlooked is the power of confirmation bias, motivated reasoning and other cognitive biases that can seriously distort the many facets of immune mental health. Inoculation theory shows promise in designing techniques to make individuals resistant to the lure of fake news, in the same way that a vaccine protects against infectious diseases.

Conceptual Background of the Domain Problem

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

In the below blog we are going to see about how we are classifying the fake news with the genuine news; we are going to use several machine learning techniques and we will plot and analyse how to identify a news as fake. I have tried using several NLP techniques and arrived at a model that will classify news is fake or genuine.

Review of Literature

The purpose of the literature review is to:

1. Identify the News basis on the content and author to tell weather it is fake or not
2. Stop the spread of fake news which will potentially spread incorrect information amount the people.

To solve this problem, we are now building a model using our machine learning technique that identifies all the Fake news, using the same the news companies can avoid the spread of fake news in all the mediums.

I have used 8 different Classification algorithms and shortlisted the best on basis on the metrics of performance and I have chosen one algorithm and build a Machine Learning model in that algorithm.

Motivation for the Problem Undertaken

Fake news is a topic that has gained a lot of attention in the past few years, and for good reasons. As social media becomes widely accessible, it becomes easier to influence millions of people by spreading misinformation. As humans, we often fail to recognize if the news we read is real or fake. A study from the University of Michigan found that human participants were able to detect fake news stories only 70 percent of the time. But can a neural network do any better? Keep reading to find out.

The goal of this article is to answer the following questions:

- What kinds of topics or keywords appear frequently in real news versus fake news?
- How can we use a deep neural network to identify fake news stories?

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

I start analysis on this project in importing the data set and simple play around with the data and identifying the characteristics of each column.

I noticed that there are four columns “id”, “headlines”, “written_by”, “news”, “labels”.

Id column only have unique variables which won’t help us in predicting.so I decided to drop ID. And I also checked there were about 109 duplicate records I dropped those records also.

```
df.drop('id',axis=1,inplace=True)
```

```
df.duplicated().sum()
```

```
109
```

```
df = df.drop_duplicates() #Droping duplicate records.
```

Post this I have checked about the null values in the data

```
df.isnull().sum()
```

```
headline        518
written_by     1932
news             39
label             0
dtype: int64
```

I changed the null values in writtern_by as “not_avaliable” and dropped the null values in the remaining data.

```
df.written_by.replace(np.nan, 'not_avaliable', inplace = True) #Filling Auther name as not_avaliable inplace of null variable.
```

```
df.dropna(subset=['headline','news'],inplace=True)   #droping headline and news null values.
```

Then post this I analysed the label column which is our target variable and I understood that label column has two variables ‘0’ and ‘1’. ‘0’denotes not a fake news and 1 denotes fake news.

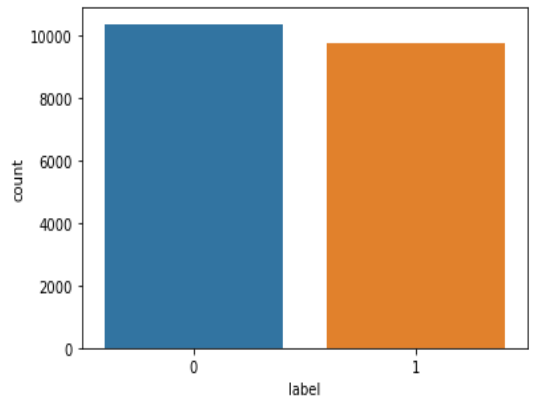
```
df.label.value_counts()
```

```
0    10387
1     9747
Name: label, dtype: int64
```

We have a balanced data it will be much helpfull for better prediction.

```
sns.countplot(df.label)
```

```
<AxesSubplot:xlabel='label', ylabel='count'>
```




```
fake_news = df[(df.label==1)]
percent=len(fake_news)/len(df)*100
print('Percentage of Fake = ',percent)
print('Percentage of not Fake news= ', (100-percent))
```

```
Percentage of Fake =  48.41064865401808
Percentage of not Fake news=  51.58935134598192
```

On further analysis of the label data, I understood that we have a balanced data almost 50% of data with fake news and not fake news. Balance data will help us in building a perfect machine learning model and we also avoid the model to overfit and underfit with the data.

Data Sources and their formats

There are 6 columns in the dataset. The description of each of the column is given below:

- “id”: Unique id of each news article
- “headline”: It is the title of the news.
- “news”: It contains the full text of the news article
- “Unnamed:0”: It is a serial number
- “written_by”: It represents the author of the news article
- “label”: It tells whether the news is fake (1) or not fake (0).

Data Pre-processing Done

I started the pre-processing with cleansing the data, filtering out all the Bash data and I like to keep only the required data for our analysis.

I started with importing the required libraries. And I have declared stop words and lemmatize to a variable.

```
#Importing Required libraries
import nltk
import re
import string
from nltk.corpus import stopwords
from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
#Defining the stop words
stop_words = stopwords.words('english')

#Defining the lemmatizer
lemmatizer = WordNetLemmatizer()
```

Then I have created a function to clean the data as like below.

```
#Function Definition for using regex operations and other text preprocessing for getting cleaned texts
def clean_comments(text):

    #convert to lower case
    lowered_text = text.lower()

    #Replacing email addresses with 'emailaddress'
    text = re.sub(r'^.+@[^\.].*\.[a-z]{2,}$', 'emailaddress', lowered_text)

    #Replace URLs with 'webaddress'
    text = re.sub(r'http\S+', 'webaddress', text)

    #Removing numbers
    text = re.sub(r'[0-9]', " ", text)

    #Removing the HTML tags
    text = re.sub(r"<.*?>", " ", text)

    #Removing Punctuations
    text = re.sub(r'[^\w\s]', ' ', text)
    text = re.sub(r'\_',' ',text)

    #Removing all the non-ascii characters
    clean_words = re.sub(r'[^\x00-\x7f]',r'', text)

    #Removing the unwanted white spaces
    text = " ".join(text.split())

    #Splitting data into words
    tokenized_text = word_tokenize(text)

    #Removing remaining tokens that are not alphabetic, Removing stop words and Lemmatizing the text
    removed_stop_text = [lemmatizer.lemmatize(word) for word in tokenized_text if word not in stop_words if word.isalpha()]

    return " ".join(removed_stop_text)
```

Then I passed my data in this function to get the data cleaned.

```
df['headline'] =df['headline'].apply(clean_comments)
df['headline']
```

```
0        ethic question dogged agriculture nominee geor...
1        u must dig deep stop argentina lionel messi ne...
2        cotton house walk plank vote bill pas senate b...
3        paul lepage besieged maine governor sends conf...
4                                      digital trump win
                               ...
20794    one police shift patrolling anxious america ne...
20796                  albert pike european migrant crisis
20797    dakota access caught infiltrating protest inci...
20798                stretch summer solstice new york time
20799    emory university pay percent undocumented stud...
Name: headline, Length: 20134, dtype: object
```

```
df['news'] = df['news'].apply(clean_comments)
df['news']
```

```
0        washington sonny perdue telling georgian growi...
1        houston venezuela plan tactical approach desig...
2        sunday abc week discussing republican plan rep...
3        augusta beleaguered republican governor maine ...
4        finian cunningham written extensively internat...
                               ...
20794    policing america today rib dinner paid strange...
20796    rixon stewart november rixon stewart nov migra...
20797    posted eddie know dakota access pipeline prote...
20798    officially summer society boutique society mem...
20799    emory university atlanta georgia announced fun...
Name: news, Length: 20134, dtype: object
```

To understand how much data, I have removed I calculated with the length of the column before cleansing and I calculated length of the columns after cleansing

```
#Checking Total Length removal in dataset
print("Original Length:", df.length_before_cleaning1.sum())
print("Cleaned Length:", df.length_after_cleaning1.sum())
print("Total Words Removed:", (df.length_before_cleaning1.sum()) - (df.length_after_cleaning1.sum()))
```

```
Original Length: 1496123
Cleaned Length: 1152116
Total Words Removed: 344007
```

```
#Checking Total Length removal in dataset
print("Original Length:", df.length_before_cleaning2.sum())
print("Cleaned Length:", df.length_after_cleaning2.sum())
print("Total Words Removed:", (df.length_before_cleaning2.sum()) - (df.length_after_cleaning2.sum()))
```

```
Original Length: 93907449
Cleaned Length: 62540918
Total Words Removed: 31366531
```

Nearly 35% of the data I have cleaned further I have split the data into X and y before training and converted the data into vectors by TFIDF vectorizer.

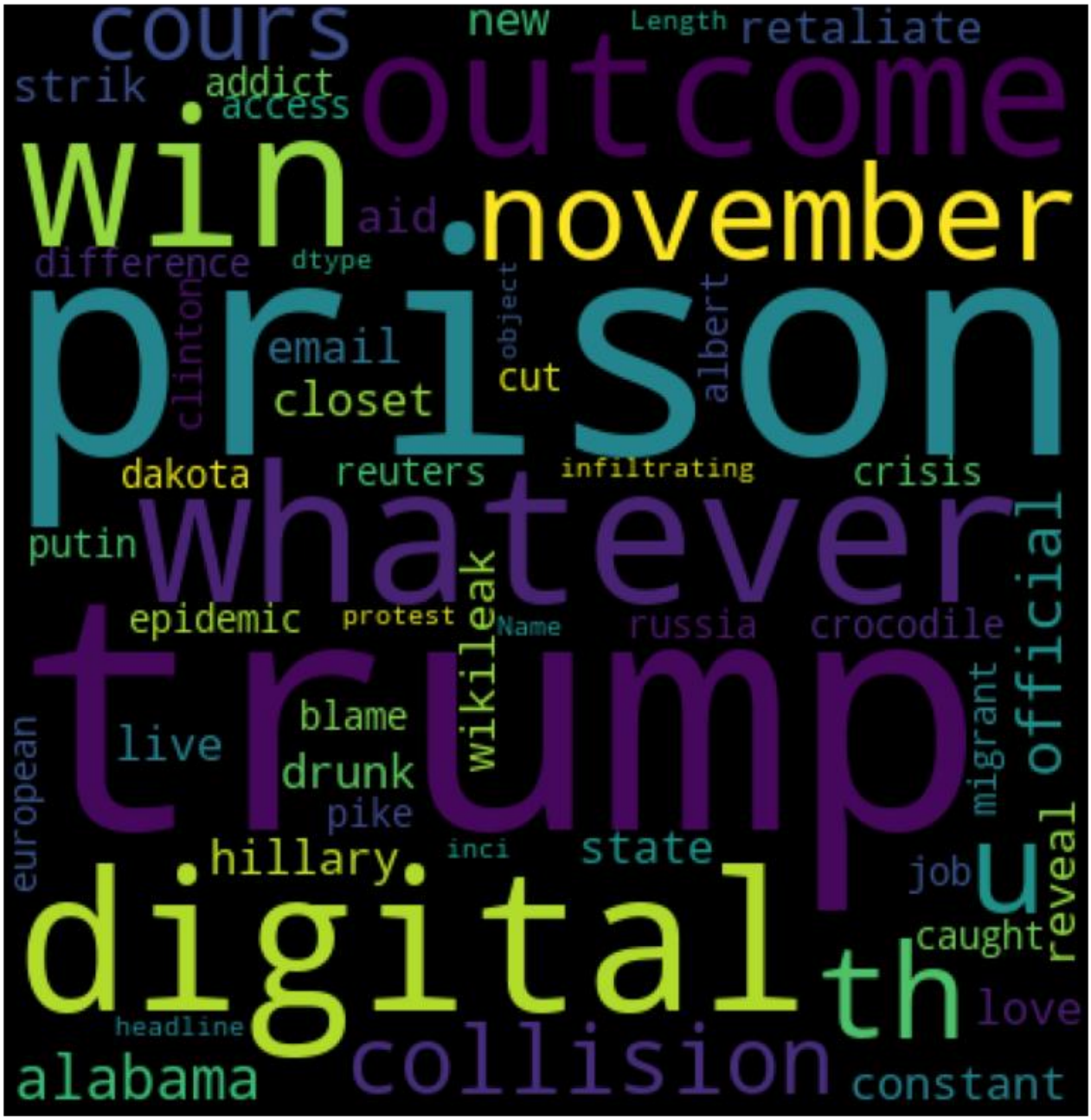
Data Inputs- Logic- Output Relationships

To understand the input and output technique I used Word Cloud Plot to understand what are the repeated words in a category.

Key Observations:

1. We can evidently see that most of the fake news are from "not_avaliable", "Dtype', "Length", "Editor" which means these fake news sources are not available.
2. News without a proper author name is being a fake news.

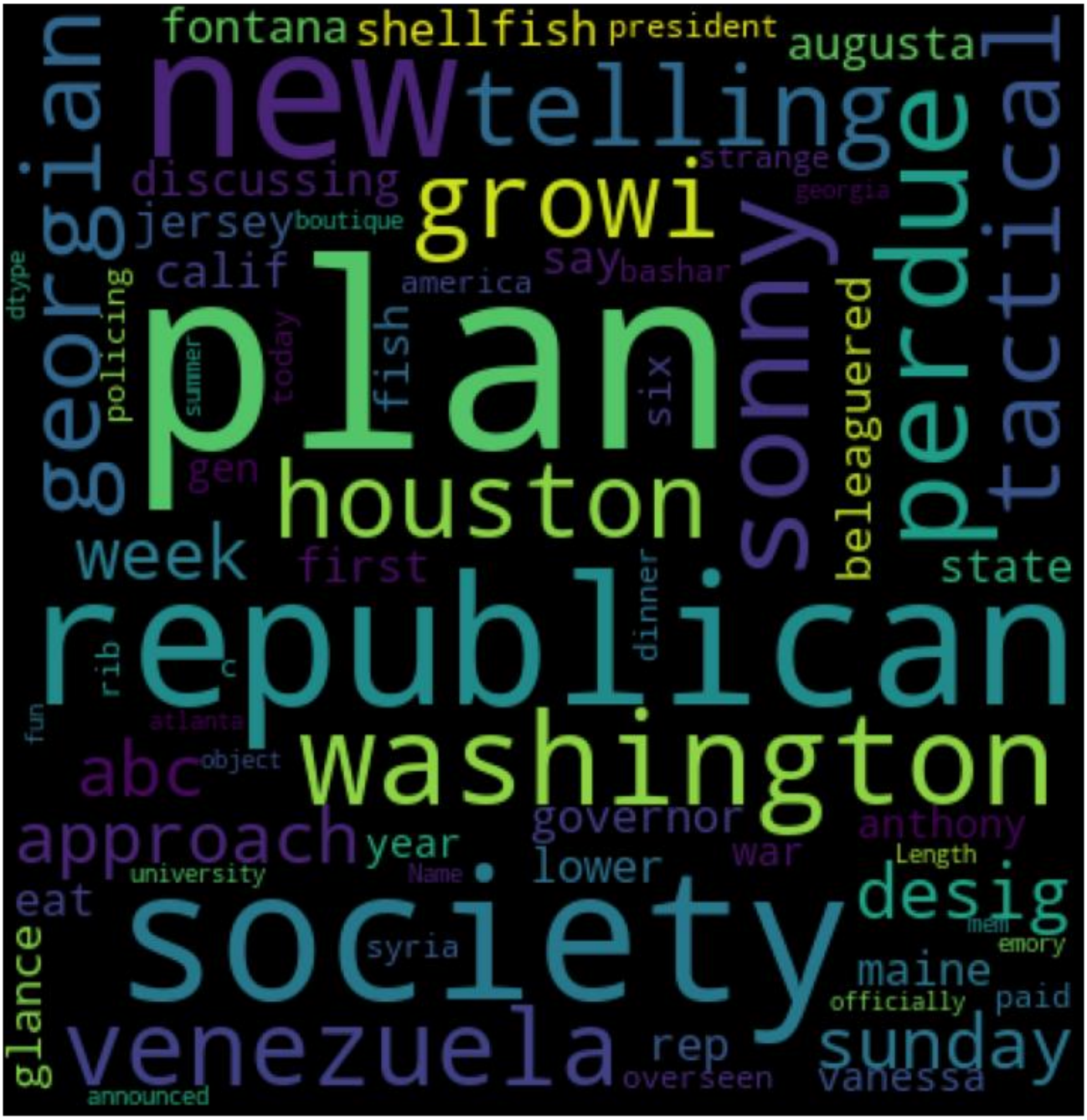
Fake News headlines



Key Observations:

1. Fake news headlines have most repeated keywords Trump, Prison, win outcome November. So, any news on these keywords might be fake.
2. This data should be taken at the time of election or before the time of election because we link the key words, we can understand these news's are fake propaganda.

Fake News Words

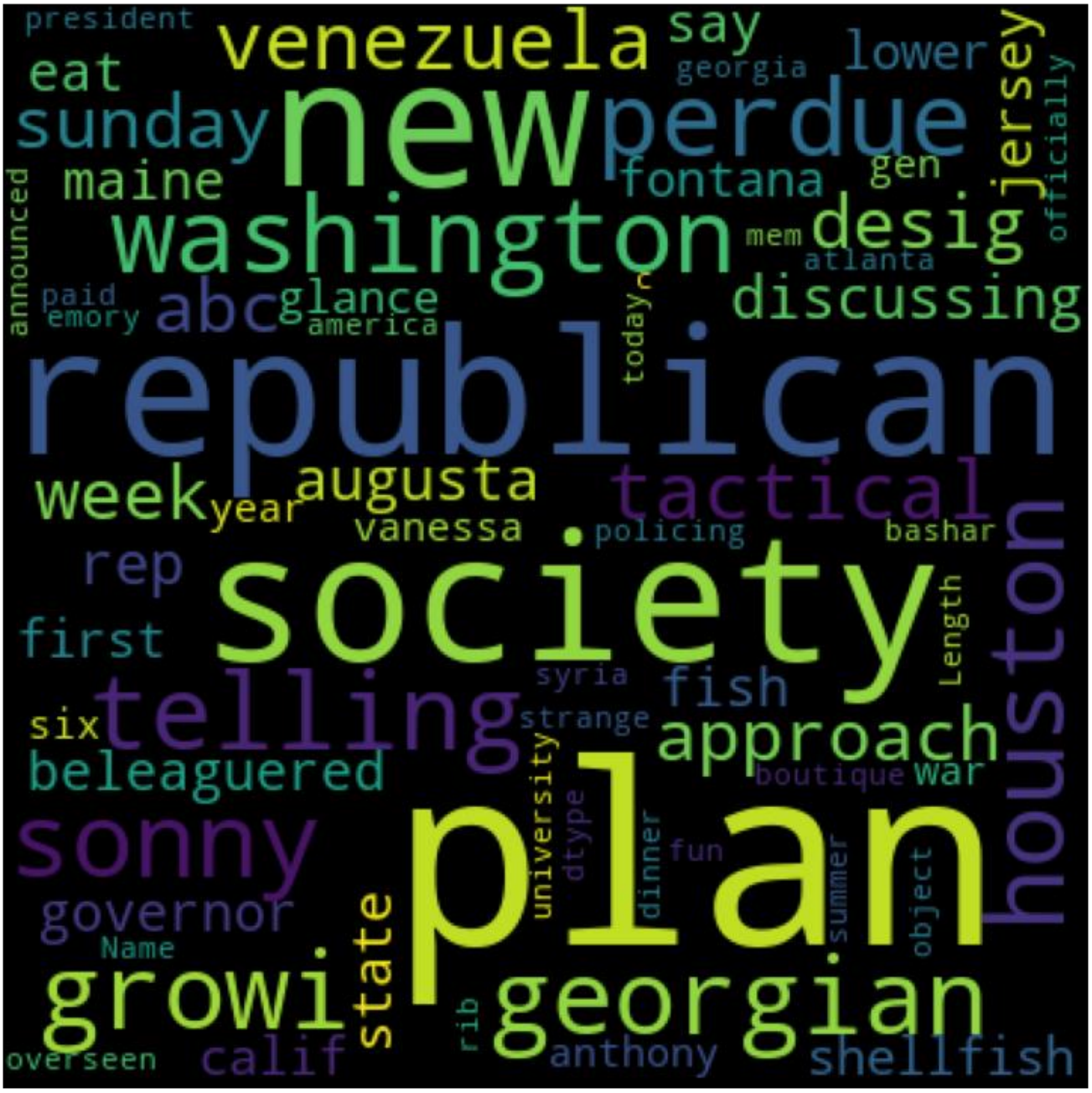


Key Observations:

1. Fake news's has most repeated keywords Washington, republican, plan, Society so any news on these keywords might be fake
2. As live we guessed before there are more chances that this data might have taken before the time of election or might be some funded individuals might have speeded this news's.

We will now see about the not Fake News highlighting words.

Not Fake News Words



Key Observations:

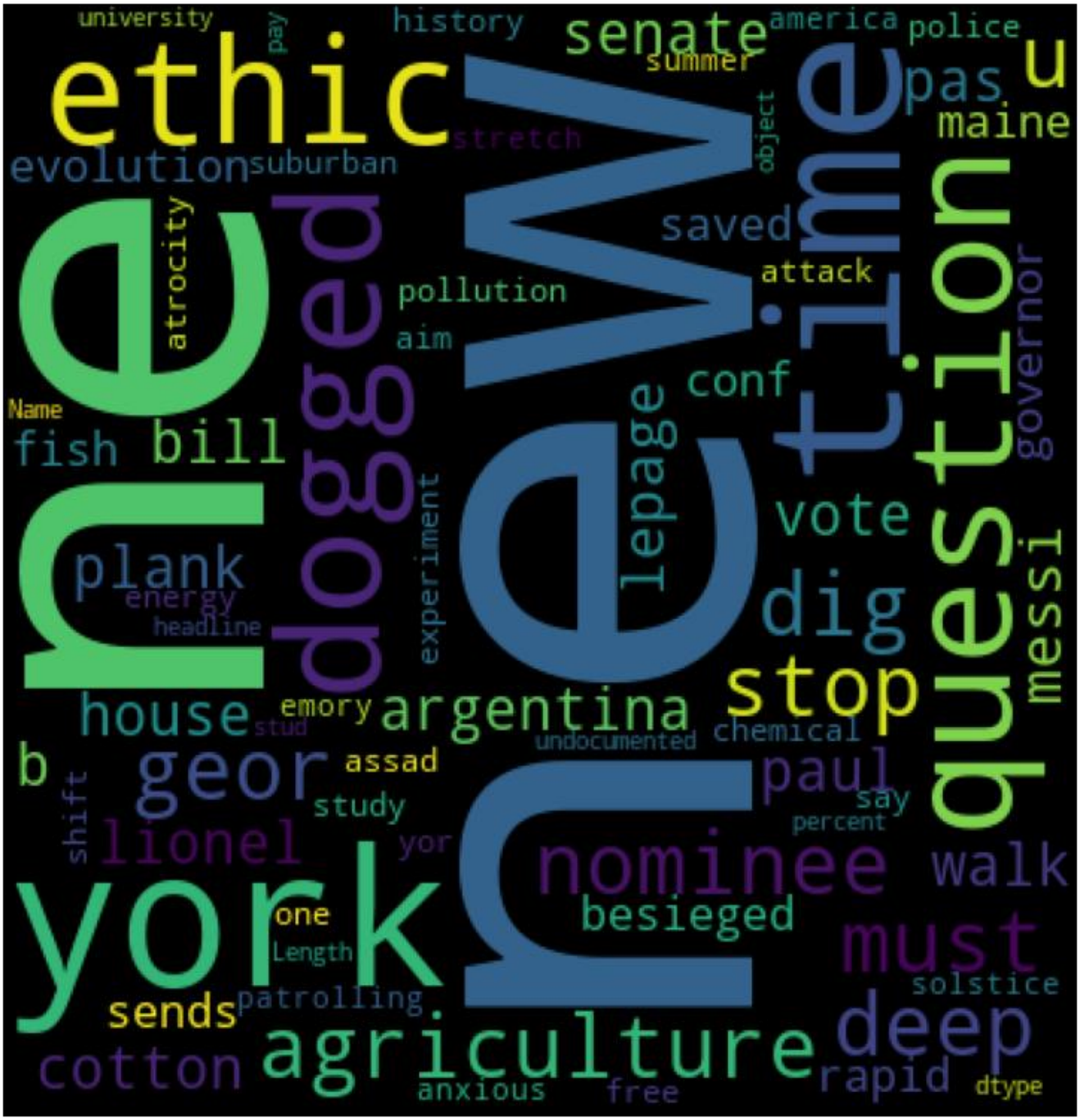
1. Not Fake news have most repeated keywords Washington, republican, plan, Society which are as same as repeated words in fake new. So, news won't be much helpfully in our prediction.
2. And we also see week, Augusta, sonny, telling, Georgian are being the most repeated words in not Fake news.

Not Fake News Authors

Key Observations:

1. In Not Fake news we can see many authors, which is a clear indication that if author names are available that possibility of being not fake is high.
2. We can see and compare with the fake news authors that there are no bogus names here. And also, our imputed name not_avaliable is also not in this list.

Not Fake News headlines



Key Observations:

1. In not fake headlines we can see New, ethic, time, culture, question, agriculture, York.
2. We will use written by and headlines in our model building.
3. And we can clearly notice the headlines difference from the fake headlines and there no words matching with fake headlines.

Hardware and Software Requirements and Tools Used

1. Python 3.8.
2. NumPy.
3. Pandas.
4. Matplotlib.
5. Seaborn. 6. Data science.
6. SciPy
7. Sklearn.
8. Anaconda Environment, Jupyter Notebook.

Model/s Development and Evaluation

Testing of Identified Approaches (Algorithms)

I have started the training in selecting the best random state parameter for the model as follows.

Training the model

```
### Selecting parameters for training
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import train_test_split

accu = 0
for i in range(0,500):
    x_train, x_test, y_train, y_test = train_test_split(X,y,test_size = .25, random_state = i)
    mod = LogisticRegression()
    mod.fit(x_train,y_train)
    y_pred = mod.predict(x_test)
    acc = accuracy_score(y_test,y_pred)
    if acc> accu:
        accu= acc
        best_rstate=i

print(f"Best Accuracy {accu*100} found on randomstate {best_rstate}")
```

```
Best Accuracy 95.87897626149952 found on randomstate 88
```

```
x_train, x_test, y_train, y_test = train_test_split(X,y,test_size = .25, random_state = best_rstate,stratify=y)
```

After selecting the best random state parameter, I have spitted the data into test and train with test size as 25 %. Again, I have imported the required libraries to import my ML algorithms.

Selecting the Best model for Training

```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import RandomizedSearchCV, cross_val_score, cross_validate, cross_val_predict
from sklearn.linear_model import SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier, ExtraTreeClassifier
from sklearn import metrics
import xgboost as xgb

import warnings
warnings.filterwarnings('ignore')
```

```
def sort_mod(estimator, x_train, y_train, cv=5, verbose=True):

    scoring = {"accuracy": "accuracy",
               "precision": "precision_weighted",
               "recall": "recall_weighted",
               "f1": "f1_weighted"}
    scores = cross_validate(estimator, x_train, y_train, cv=cv, scoring=scoring)
    accuracy, accuracy_std = scores['test_accuracy'].mean(),scores['test_accuracy'].std()
    precision, precision_std = scores['test_precision'].mean(),scores['test_precision'].std()
    recall, recall_std = scores['test_recall'].mean(),scores['test_recall'].std()
    f1, f1_std = scores['test_f1'].mean(),scores['test_f1'].std()


    ScoRes = {"Accuracy": accuracy,"Accuracy std": accuracy_std,"Precision": precision,"Precision std": precision_std,
              "Recall": recall,"Recall std": recall_std,"f1": f1,"f1 std": f1_std,}

    if verbose:
        print(f"Accuracy: {accuracy} - (std: {accuracy_std})")
        print(f"Precision: {precision} - (std: {precision_std})")
        print(f"Recall: {recall} - (std: {recall_std})")
        print(f"f1: {f1} - (std: {f1_std})")

    return ScoRes
```

Run and evaluate selected models

As we can see above, I have imported 8 classification algorithms and I am going to shortlist the best amongst these in basis of accuracy precision recall and F1 scores.

```
models = [LogisticRegression(), RandomForestClassifier(random_state=42),
          DecisionTreeClassifier(random_state=42), ExtraTreeClassifier(random_state=42),
          AdaBoostClassifier(random_state=42), GradientBoostingClassifier(random_state=42),
          xgb.XGBClassifier()]

model_names = ["LogisticRegression", "Random Forest",
               "Decision Tree", "Extra Tree", "Ada Boost",
               "Gradient Boosting", "XGBoost"]
```

```
accuracy = []
precision = []
recall = []
f1 = []

for model in range(len(models)):
    print(f"\n\nStep {model+1} of {len(models)}")
    print(f".....................................................................running {model_names[model]}")

    clf_scores = sort_mod(models[model], x_train, y_train)

    accuracy.append(clf_scores["Accuracy"])
    precision.append(clf_scores["Precision"])
    recall.append(clf_scores["Recall"])
    f1.append(clf_scores["f1"])
```

So, like above I have run all the algorithms with the data.

Key-Metrics for success in solving problem under consideration.

I have taken key metrics as Accuracy, Precision, Recall and F1 scores to analysis the best model.

	Model	accuracy	precision	recall	f1
1	Random Forest	0.992914	0.992953	0.992914	0.992915
2	Decision Tree	0.992053	0.992060	0.992053	0.992053
6	XGBoost	0.988742	0.988852	0.988742	0.988744
4	Ada Boost	0.980066	0.980287	0.980066	0.980071
0	LogisticRegression	0.979868	0.980230	0.979868	0.979872
5	Gradient Boosting	0.965232	0.967185	0.965232	0.965231
3	Extra Tree	0.939338	0.939466	0.939338	0.939341

As we can see above Random Forest tops the chart, I have selected Random Forest model as my final model and I have Hyper parameter tuned the same to increase the performance of the model and have achieved the accuracy of 99.364 % and I have saved the model.

```
preds = cross_val_predict(clf_random.best_estimator_, x_train, y_train, cv=5, n_jobs=-1)
print(metrics.classification_report(y_train, preds, zero_division=0))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      7790
           1       0.99      1.00      0.99      7310

    accuracy                           0.99     15100
   macro avg       0.99      0.99      0.99     15100
weighted avg       0.99      0.99      0.99     15100
```

CONCLUSION

Key Findings and Conclusions of the Study

The finding of the study is that when the news's are being published on a bogus name, the author names not available that news are end up being Fake, and also, we can understand this fake news's are desperately being spread among the public to create a fake image of an individual, or to get profit out of it or to destroy the good deeds of the target person.

Learning Outcomes of the Study in respect of Data Science

The universe of "fake news" is much larger than simply false news stories. Some stories may have a nugget of truth, but lack any contextualizing details. They may not include any verifiable facts or sources. Some stories may include basic verifiable facts, but are written using language that is deliberately inflammatory, leaves out pertinent details or only presents one viewpoint. "Fake news" exists within a larger ecosystem of mis- and disinformation.

Misinformation is false or inaccurate information that is mistakenly or inadvertently created or spread; the intent is not to deceive. Disinformation is false information that is deliberately created and spread "in order to influence public opinion or obscure the truth"

-(https://www.merriam-webster.com/dictionary/disinformation).

As per our evaluation, we found that lesser number of Authors or bogus names or authors unknown have released fake news. We trained 20800 observations for five context categories using a Random Forest algorithm for context detection. Then, the system classifies the fake news in one of the trained contexts in the text conversation. In our testbed, we observed 48.41% of records have fake news but if we

search for the authors names in fake news only 10% of the authors spread almost all the fake news. Hence, our proposed approach can identify the Fake news and the authors who spread fake news, as discussed usually on a no source news or on a bogus name these fake news's are spread.

Limitations of this work and Scope for Future Work

The limitation of the study is that this data was taken in a shorter time frame on a current trend which might help us in a prediction for a shorted period of time. So, if the prediction of fake news was done with very old data with our model there are chances that the prediction won't be accurate. Same applies for not immediate future data. So, in such case if we have analysis the trend of the news, and if we split the news category as politics, sports arts, general, local, international then we might get some accurate prediction.