FLIP ROBO

Fake News Detection

Submitted by:

Aditi Gupta

Internship 12

ACKNOWLEDGMENT

Gratitude takes three forms-"A feeling from heart, an expression in words and a giving in return". We take this opportunity to express our feelings.

I express my gracious gratitude to our project guide **Khushboo Garg**, SME of Flip Robo Technologies, for his valuable guidance and assistance. I am very thankful for her encouragement and inspiration that made project successful.

I express my gracious gratitude to our Trainer **Dr. Deepika Sharma,** Training Head of DataTrained - Data Analytics, Data Science Online Training, Bengaluru, for her valuable guidance and assistance throughout the PG Program. I am very thankful for her encouragement and inspiration that made project successful.

I would like to thank **Shankar**, In House Data Scientist of DataTrained – Data Analytics, Data Science Online Training, Bengaluru for his Profound guidance throughout the training.

My special thanks to all the **Instructors and Subject Matter Experts** of DataTrained – Data Analytics, Data Science Online Training, Bengaluru, Who helped me during the live sessions and during doubt clearing Scenarios from which I received a lots of suggestions that improved the quality of the work.

I express my deep sense of gratitude to my family for their moral support and understanding without which the completion of my project would not have been perceivable.

INTRODUCTION

- Business Problem Framing

  Fake news is a form of news consisting of delibrate disinformation or hoaxes spread via traditional news media or online social media. In this project, I have different natural language processing (NLP) based machine learning approaches to detect fake news fron news headlines and news.

- Conceptual Background of the Domain Problem

  The idea of fake news is often referred to as clickbait in social trends and is defined as a “made up story with an intention to deceive, geared towards getting clicks”. Some news articles have titles which grab a reader’s interest. Yet, the author only emphasizes a specific part of the article in the title. If the article itself does not focus on or give much truth to what the title had written, the new may be misleading.

- Review of Literature

  If we look at some scholar work shows the issue that the fake news has been concerned amongst scholar from various background. For instance, some authors have observed that fake news is no longer a preserve of the marketing and public relation departments. Instead there is an increasing risk of IT security, therefore IT department is premised on the idea that it would help avert the various risks associated with the problem. So, if we go deeply into it we could find that the hackers use clickbait with the help of fake news and make some professional of the organization download the malicious exploits in their system or leak sensitive information, albeit in an indirect manner. The user may be tricked into believing that they are helping to disseminate the news further when in the actual sence they are providing the perpetrators with access to their emails. So we need to implement more our research and extensive knowledge to solve the problem.

- ## Motivation for the Problem Undertaken

  This project is highly motivated project as it includes the real time problem of fake news which if we see is getting bigger, as there various concern as people do good things work hard to build a reputation and only one fake news is enough to ruin it all, it also have inverse effect on financial market as we observe there will be a good amount of fluctuation on stock market based on these news only.

- ## PROBLEM STATEMENT
  - Fake news irritating internet connection
  - Critical news are missed and / or delayed.
  - Millions of compromised computers
  - Billions of dollars lost worldwide
  - Identity theft
  - Fake news can crash mail servers and fill up hard drives

- ## OBJECTIVE

  The objective of identification of fake news are :
  - To give knowledge to the user about the fake news and relevant news
  - To classify that newsis fake or not.

- SCOPE OF THE PROJECT:
  - It provides sensitivity to the client and adapts well to the future fake news techniques.
  - It considers a complete news instead of single words with respect to its organization.
  - It increases Security and Control.
  - It reduces IT Administration Costs.
  - It also reduce Network Resource Costs.

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

  Machine Learning is defined by Tom Mitchell in his book as "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". Supervised learning is when the output is known for the corresponding inputs, and is also provided for the machine to learn.

  - Data Preprocessing
  - EDA (Exploratory data analysis)
  - Building Word Dictionary
  - Feature Extraction
  - Scoring & Metrics

- ## Data Sources and their formats

  The data is provided to us from our client database. It is hereby given to us for the exercise to improve the selection of mails for spam or not spam. It is given in the csv file format.

```
# Loading the dataset
df_news=pd.read_csv('train_news.csv')
df_news
```

	Unnamed: 0	id	headline	written_by	news	label
0	0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0
2	2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0
3	3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0
4	4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1
...	...	...	...	...	...	...
20795	20795	5671	NaN	NeverSurrender	No, you'll be a dog licking of the vomit of yo...	1
20796	20796	14831	Albert Pike and the European Migrant Crisis	Rixon Stewart	By Rixon Stewart on November 5, 2016 Rixon Ste...	1
20797	20797	18142	Dakota Access Caught Infiltrating Protests to ...	Eddy Lavine	posted by Eddie You know the Dakota Access Pip...	1
20798	20798	12139	How to Stretch the Summer Solstice - The New Y...	Alison S. Cohn	It's officially summer, and the Society Boutiq...	0
20799	20799	15660	Emory University to Pay for '100 Percent' of U...	Tom Ciccotta	Emory University in Atlanta, Georgia, has anno...	0

20800 rows × 6 columns

- Data Preprocessing Done

The dataset that will be used to train the model has some challenges. Text Cleaning is a very important step in machine learning because your data may contains a lot of noise and unwanted character such as punctuation, white space, numbers, hyperlink and etc.

Some standard procedures are:

  - Checking and then Filling and dropping for the null values

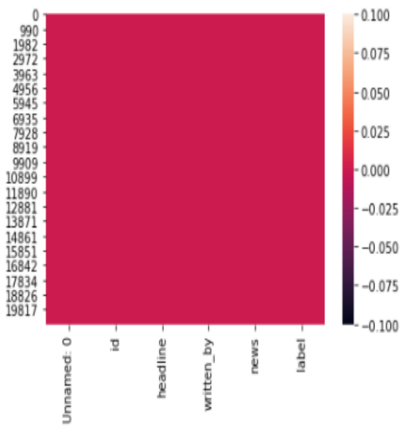
```
Dataset contains any NaN/Empty cells :  True

Total number of empty rows in each feature:
 Unnamed: 0        0
id                0
headline        558
written_by     1957
news             39
label             0
dtype: int64
```

```
# Feature subject is having some null values, in here I am
# filling 'written_by' feature with unknown because sometimes there are anonymus authors,...
# filling up empty values in 'headline' feature as i will merge it further for further processing...
# Dropping empty values in rows because we are detecting fake news here and for this news is needed..

df_news['written_by'].fillna('Unknown ',inplace=True)
df_news['headline'].fillna('No Headline ',inplace=True)
df_news.dropna(subset=['news'],inplace=True)
df_news.head()
```

	Unnamed: 0	id	headline	written_by	news	label
0	0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0
2	2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0
3	3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0
4	4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1

```
# Hetmap for null values
sns.heatmap(df_news.isnull())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a64980ed90>
```



- Adding two new column ‘Content’ by merging headlines and news Taking the length in new column ‘content_length’

```
# As we all know that title and text both are important for authenticity detection,
# thus here I am joining both title and text features to get some useful information while training...
df_news['Content'] = df_news[['headline', 'news']].apply(lambda x: ' '.join(x), axis = 1)
df_news
```

```
# New feature (length) contains length of the content feature..
df_news['content_length'] = df_news.Content.str.len()
df_news.head()
```

	id	headline	written_by	news	label	Content	content_length
0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0	Ethics Questions Dogged Agriculture Nominee as...	8021
1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0	U.S. Must Dig Deep to Stop Argentina's Lionel ...	6185
2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0	Cotton to House: 'Do Not Walk the Plank and Vo...	526
3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0	Paul LePage, Besieged Maine Governor, Sends Co...	6617
4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1	A Digital 9/11 If Trump Wins Finian Cunningh...	9193

Make a function for the following:

- Removing pos tags using wordnet
  (# Return the wordnet object value corresponding to the POS tag)
  (#Part-Of-Speech (POS) tagging: assign a tag to every word to define if it corresponds to a noun, a verb etc. using the WordNet lexical database)
- convert all letters to lower/upper case
- removing numbers
- removing punctuation
- removing white spaces
- removing hyperlink
- removing stop words such as *a, about, above, down, doing* and the list goes on... Sometimes, the extremely common word which would appear to be of very little value in helping select documents matching user need are excluded from the vocabulary entirely.
- Word lemmatizing: Lemmatizing is utilizing the dictionary of a particular language and tried to convert the words back to its base form. It will try to take into account of the meaning of the verbs and convert it back to the most suitable base form.

```python
# Return the wordnet object value corresponding to the POS tag
#Part-Of-Speech (POS) tagging: assign a tag to every word to define if it corresponds to a noun, a verb etc. using the WordNet le

def get_wordnet_pos(pos_tag):
    if pos_tag.startswith('J'):
        return wordnet.ADJ
    elif pos_tag.startswith('V'):
        return wordnet.VERB
    elif pos_tag.startswith('N'):
        return wordnet.NOUN
    elif pos_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN


def clean_text(text):
    # lower text
    text = text.lower()
    text = re.sub("[^\w\s]", " ", text)
    # tokenize text and remove puncutation
    text = [word.strip(string.punctuation) for word in text.split(" ")]
    # remove words that contain numbers
    text = [word for word in text if not any(c.isdigit() for c in word)]
    # Remove leading and trailing whitespace
    #text=re.sub("[^\s+|\s+?$]"," ",text)
    # remove stop words
    stop = set(stopwords.words('english') + ['u', 'ü', 'ur', '4', '2', 'im', 'dont', 'doin', 'ure'])
    text = [x for x in text if x not in stop]
    # remove empty tokens
    text = [t for t in text if len(t) > 0]
    # pos tag text
    pos_tags = pos_tag(text)
    # lemmatize text
    text = [WordNetLemmatizer().lemmatize(t[0], get_wordnet_pos(t[1])) for t in pos_tags]
    #text=stemmer.stem(text)
    # remove words with only two letter
    text = [t for t in text if len(t) > 2]
    # join all
    text = " ".join(text)
    return(text)
```

- Adding two new column 'clean_content' by using the above function on content column and then 'clean_column_length' which contains the length of the column.

```python
# cleaning the news and storing them in a separate feature...
df_news["clean_content"] = df_news["Content"].apply(lambda x: clean_text(x))
```

```python
# New feature (Clean_length) contains length of the Clean_content feature after puncuations, stopwords removal..
df_news['clean_content_length'] = df_news.clean_content.str.len()
df_news.head()
```

	id	headline	written_by	news	label	Content	content_length	clean_content	clean_content_length
0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0	Ethics Questions Dogged Agriculture Nominee as...	8021	ethic question dog agriculture nominee georgia...	4983
1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0	U.S. Must Dig Deep to Stop Argentina's Lionel ...	6185	must dig deep stop argentina lionel messi new ...	3886
2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0	Cotton to House: 'Do Not Walk the Plank and Vo...	526	cotton house walk plank vote bill cannot pass ...	315
3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0	Paul LePage, Besieged Maine Governor, Sends Co...	6617	paul lepage besiege maine governor send confi...	4166
4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1	A Digital 9/11 If Trump Wins Finian Cunningh...	9193	digital trump win finian cunningham write exte...	6410

Change in date Before and After doing preprocessing

Before Preprocessing

	Not Fake Words	Fake Words
0	(the, 477787)	(the, 338855)
1	(to, 245507)	(of, 178429)
2	(of, 238936)	(to, 177382)
3	(a, 221848)	(and, 157269)
4	(and, 205457)	(a, 123308)
5	(in, 173291)	(in, 108566)
6	(that, 114656)	(that, 81347)
7	(for, 82760)	(is, 78508)
8	(on, 77853)	(for, 54776)
9	(is, 72281)	(on, 45011)

After Preprocessing

	Not Fake Words	Fake Words
0	(say, 82086)	(trump, 21708)
1	(trump, 38454)	(clinton, 21164)
2	(new, 26473)	(say, 20839)
3	(time, 24583)	(people, 17072)
4	(one, 23602)	(one, 17054)
5	(state, 23552)	(state, 16670)
6	(would, 22870)	(would, 14610)
7	(year, 21785)	(hillary, 13814)
8	(people, 20124)	(make, 13081)
9	(make, 19015)	(time, 12910)

Building Word Dictionary

```
# Tokenizing Documents..
data=[]
from nltk.tokenize import word_tokenize
for j,i in enumerate(df_news['clean_content']):
    a=word_tokenize(i,'english')
    data.append(a)
```

```
# Making Word dictionary...
dictionary = corpora.Dictionary(data)
print(dictionary)
```

```
Dictionary(163632 unique tokens: ['acceptable', 'accompany', 'accord', 'act', 'action']...)
```

- Hardware and Software Requirements and Tools Used

  Hardware : Since the computational aspect of the project is of importance to PANDA, it is important to know the hardware that was used in the evaluation process. The training and evaluation of the neural network model has been done on a Windows 10 computer using a quad-core CPU at i3.

  Software : anaconda 3 , windows 10 ,Microsoft office.

  Tools used : python , machine learning libraries, Nltk, Nlp libraries.

- Feature Extraction

```
# creating the TF-IDF(term frequency-inverse document frequency) vectorizer function in order to convert the tokens
#   from the train documents into vectors so that machine can do further processing
def Tf_idf_train(text):
    tfid = TfidfVectorizer(min_df=3,smooth_idf=False)
    return tfid.fit_transform(text)
```

```
# Inserting vectorized values in a variable x, which will be used in training the model
x=Tf_idf_train(df_news['clean_content'])

# checking the shape of the data which is inserted in x which will be used for model training.
print("Shape of x: ",x.shape)
```

```
Shape of x:  (20761, 54349)
```

```
# Assigning the label in y and checking it's shape
y = df_news['label'].values
print("Shape of y: ",y.shape)
```

```
Shape of y:  (20761,)
```

Model/s Development and Evaluation

- Models Used

```
# Importing useful libraries for model training

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier

# Ensemble Techniques...

from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import AdaBoostClassifier

# Model selection libraries...
from sklearn.model_selection import cross_val_score, cross_val_predict, train_test_split
from sklearn.model_selection import GridSearchCV

# Importing some metrics we can use to evaluate our model performance....
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import precision_score, recall_score, f1_score

# Creating instances for different Classifiers

RF=RandomForestClassifier()
LR=LogisticRegression()
MNB=MultinomialNB()
DT=DecisionTreeClassifier()
AD=AdaBoostClassifier()
XG=XGBClassifier()
```

```
Model = []
score = []
cvs=[]
rocscore=[]
for name,model in models:
    print('***********************',name,'***********************')
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train)
    print(model)
    pre=model.predict(x_test)
    print('\n')
    AS=accuracy_score(y_test,pre)
    print('Accuracy_score = ',AS)
    score.append(AS*100)
    print('\n')
    sc = cross_val_score(model, x, y, cv=10, scoring='accuracy').mean()
    print('Cross_Val_Score = ',sc)
    cvs.append(sc*100)
    print('\n')
    false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test,pre)
    roc_auc = auc(false_positive_rate, true_positive_rate)
    print ('roc_auc_score = ',roc_auc)
    rocscore.append(roc_auc*100)
    print('\n')
    print('classification_report\n',classification_report(y_test,pre))
    print('\n')
    cm=confusion_matrix(y_test,pre)
    print(cm)
    print('\n')
    plt.figure(figsize=(10,40))
    plt.subplot(911)
    plt.title(name)
    print(sns.heatmap(cm,annot=True))
    plt.subplot(912)
    plt.title(name)
    plt.plot(false_positive_rate, true_positive_rate, label='AUC = %0.2f'% roc_auc)
    plt.plot([0,1],[0,1],'r--')
    plt.legend(loc='lower right')
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    print('\n\n')
```

- Run and Evaluate selected models

  When it comes to evaluation of a data science model's performance, sometimes accuracy may not be the best indicator. Some problems that we are solving in real life might have a very imbalanced class and using accuracy might not give us enough confidence to understand the algorithm's performance.

  In the fake news detection problem that we are trying to solve, the fake news is approximately 50% of our data. If our algorithm predicts all the email as non-spam, it will achieve an accuracy of 50%. And for some problem that has only 1% of positive data, predicting all the sample as negative will give them an accuracy of 99% but we all know this kind of model is useless in a real life scenario.

  Precision & Recall is the common evaluation metrics that people use when they are evaluating class-imbalanced classification model.Let's try to understand what questions Precision & Recall is trying to answer,

  - Precision: What proportion of positive identifications was actually correct ?
  - Recall: What actual proportion of actual positives was identified correctly?

  So, precision is evaluating, when a model predict something as positive, how accurate the model is. On the other hand, recall is evaluating how well a model in finding all the positive samples.

  Confusion Matrix

  Confusion Matrix is a very good way to understand results like true positive, false positive, true negative and so on.

  Sklearn documentation has provided a sample code of how to plot nice looking confusion matrix to visualize your result..

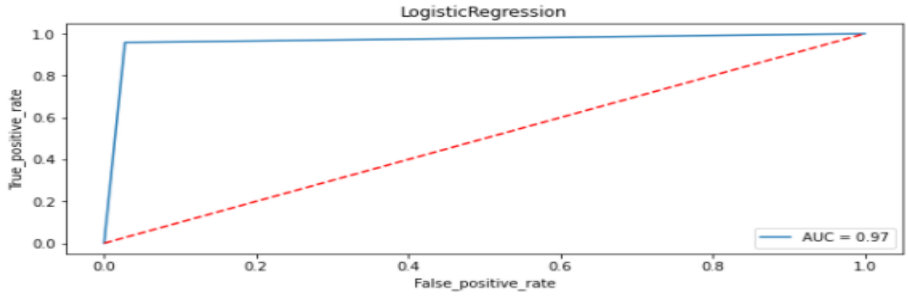
  Confusion Matrix of the result

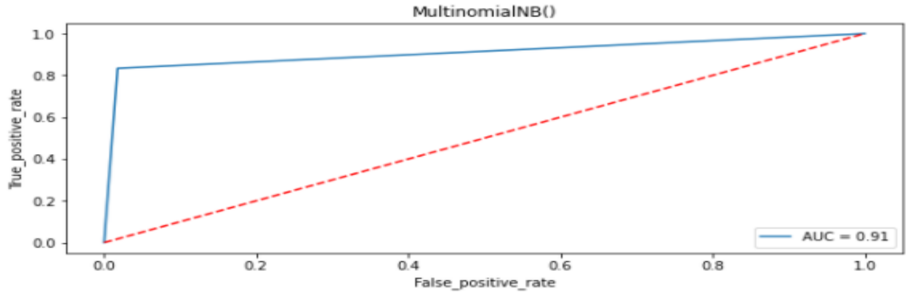
```
*************************** LogisticRegression *****************************


LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)


Max Accuracy Score corresponding to Random State  61 is: 0.9685342751645529


Learning Score :  0.9832782824112304
Accuracy Score :  0.9654840263284636
Cross Val Score :  0.9947618843675959
roc auc score :  0.965480538979566


Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.97      0.97      3116
           1       0.97      0.96      0.97      3113

    accuracy                           0.97      6229
   macro avg       0.97      0.97      0.97      6229
weighted avg       0.97      0.97      0.97      6229



Confusion Matrix:
 [[3031   85]
 [ 130 2983]]

*************************** MultinomialNB() *****************************


MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)


Max Accuracy Score corresponding to Random State  71 is: 0.9181248996628673


Learning Score :  0.9364849986237269
Accuracy Score :  0.9081714560924707
Cross Val Score :  0.9843101497880349
roc auc score :  0.9081357135405091


Classification Report:
               precision    recall  f1-score   support

           0       0.86      0.98      0.91      3116
           1       0.98      0.83      0.90      3113

    accuracy                           0.91      6229
   macro avg       0.92      0.91      0.91      6229
weighted avg       0.92      0.91      0.91      6229



Confusion Matrix:
 [[3061   55]
 [ 517 2596]]
```

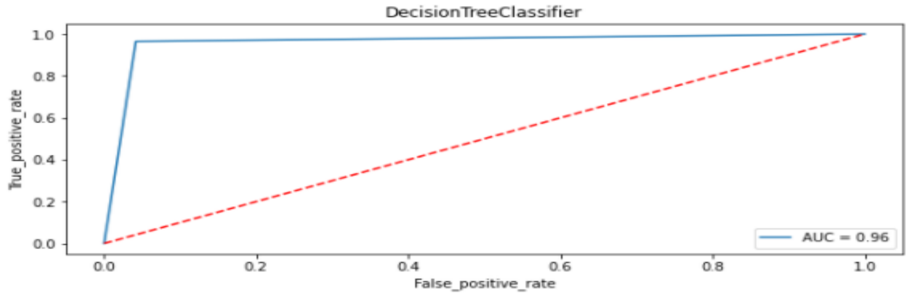
```
*************************** DecisionTreeClassifier *****************************


DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')


Max Accuracy Score corresponding to Random State  92 is: 0.9643602504414834


Learning Score :  1.0
Accuracy Score :  0.9616310804302456
Cross Val Score :  0.9597815499959201
roc auc score :  0.9616326952235996


Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.96      0.96      3116
           1       0.96      0.96      0.96      3113

    accuracy                           0.96      6229
   macro avg       0.96      0.96      0.96      6229
weighted avg       0.96      0.96      0.96      6229



Confusion Matrix:
 [[2986  130]
 [ 109 3004]]


 *************************** RandomForestClassifier *****************************


 RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)


 Max Accuracy Score corresponding to Random State  43 is: 0.9717450634130679


 Learning Score :  1.0
 Accuracy Score :  0.9677315781024242
 Cross Val Score :  0.9954496494545729
 roc auc score :  0.9677273180875925


 Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.98      0.97      3116
           1       0.98      0.96      0.97      3113

    accuracy                           0.97      6229
   macro avg       0.97      0.97      0.97      6229
weighted avg       0.97      0.97      0.97      6229



 Confusion Matrix:
 [[3043   73]
 [ 128 2985]]
```

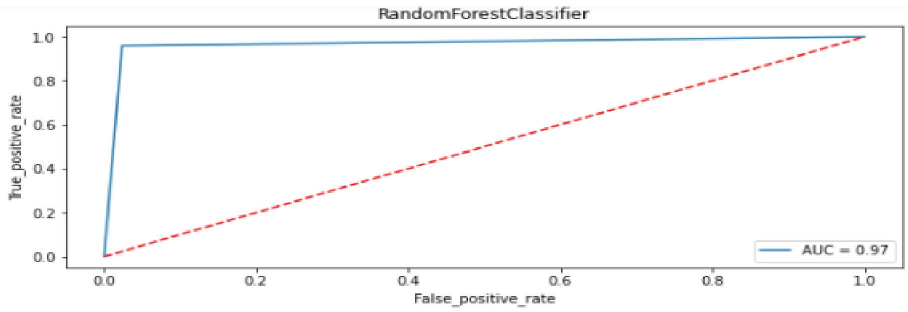
```
*************************** AdaBoostClassifier *****************************


AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None, learning_rate=1.0,
                   n_estimators=50, random_state=None)


Max Accuracy Score corresponding to Random State  97 is: 0.9749558516615829


Learning Score :  0.9761904761904762
Accuracy Score :  0.9693369722266817
Cross Val Score :  0.99569414839702
roc auc score :  0.9693371970703831


Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.97      0.97      3116
           1       0.97      0.97      0.97      3113

    accuracy                           0.97      6229
   macro avg       0.97      0.97      0.97      6229
weighted avg       0.97      0.97      0.97      6229



Confusion Matrix:
 [[3019   97]
 [  94 3019]]
```

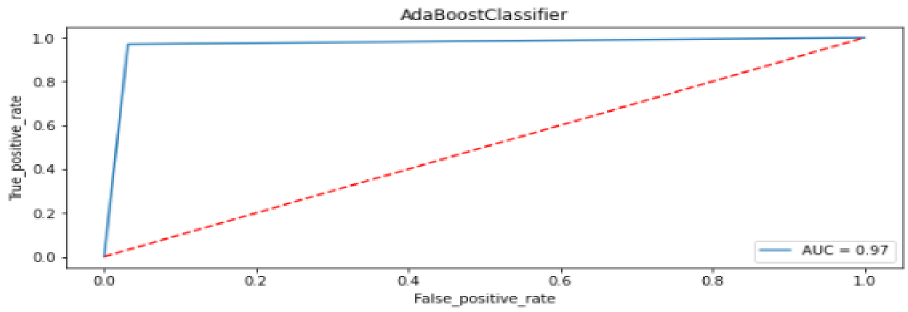
```
*************************** XGBClassifier *****************************


XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)


Max Accuracy Score corresponding to Random State  44 is: 0.9735109969497512


Learning Score :  0.9755023396641894
Accuracy Score :  0.9719056028254937
Cross Val Score :  0.9964623358557286
roc auc score :  0.9719089210140752


Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.97      0.97      3116
           1       0.97      0.98      0.97      3113

    accuracy                           0.97      6229
   macro avg       0.97      0.97      0.97      6229
weighted avg       0.97      0.97      0.97      6229



Confusion Matrix:
 [[3007  109]
 [  66 3047]]
```

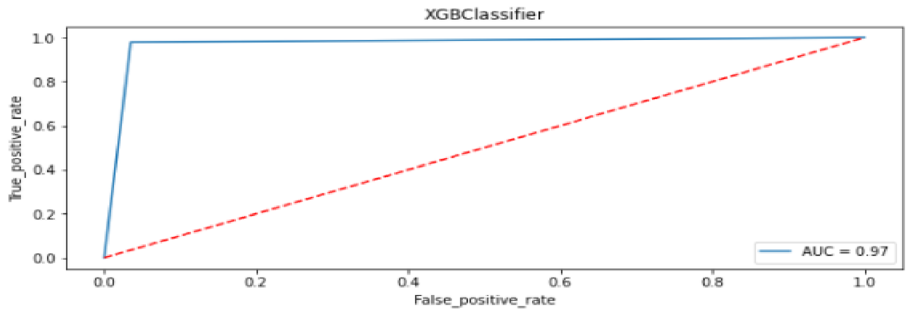

LogisticRegression
True_positive_rate
False_positive_rate
AUC = 0.97


MultinomialNB()
True_positive_rate
False_positive_rate
AUC = 0.91


DecisionTreeClassifier
True_positive_rate
False_positive_rate
AUC = 0.96


RandomForestClassifier
True_positive_rate
False_positive_rate
AUC = 0.97


AdaBoostClassifier
True_positive_rate
False_positive_rate
AUC = 0.97


XGBClassifier
True_positive_rate
False_positive_rate
AUC = 0.97

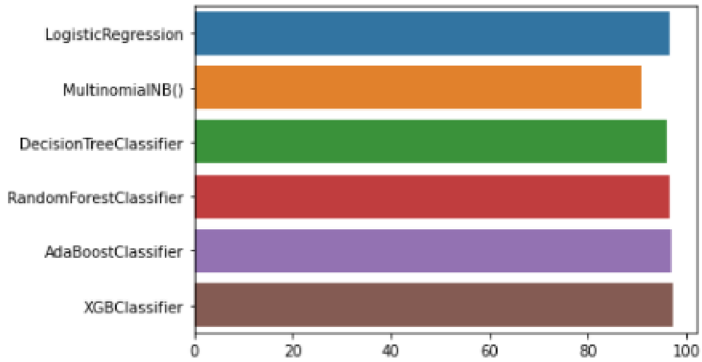
- Key Metrics for success in solving problem under consideration

```python
# Making a Dataframe comprises of Differnt Calculated Scores :
result=pd.DataFrame({'Model': Model,'Learning Score': Score,'Accuracy Score': Acc_score,'Cross Val Score':cvs,
                     'Roc_Auc_curve':rocscore})
result
```

	Model	Learning Score	Accuracy Score	Cross Val Score	Roc_Auc_curve
0	LogisticRegression	98.327828	96.548403	99.476188	96.548054
1	MultinomialNB()	93.648500	90.817146	98.431015	90.813571
2	DecisionTreeClassifier	100.000000	96.163108	95.978155	96.163270
3	RandomForestClassifier	100.000000	96.773158	99.544965	96.772732
4	AdaBoostClassifier	97.619048	96.933697	99.569415	96.933720
5	XGBClassifier	97.550234	97.190560	99.646234	97.190892

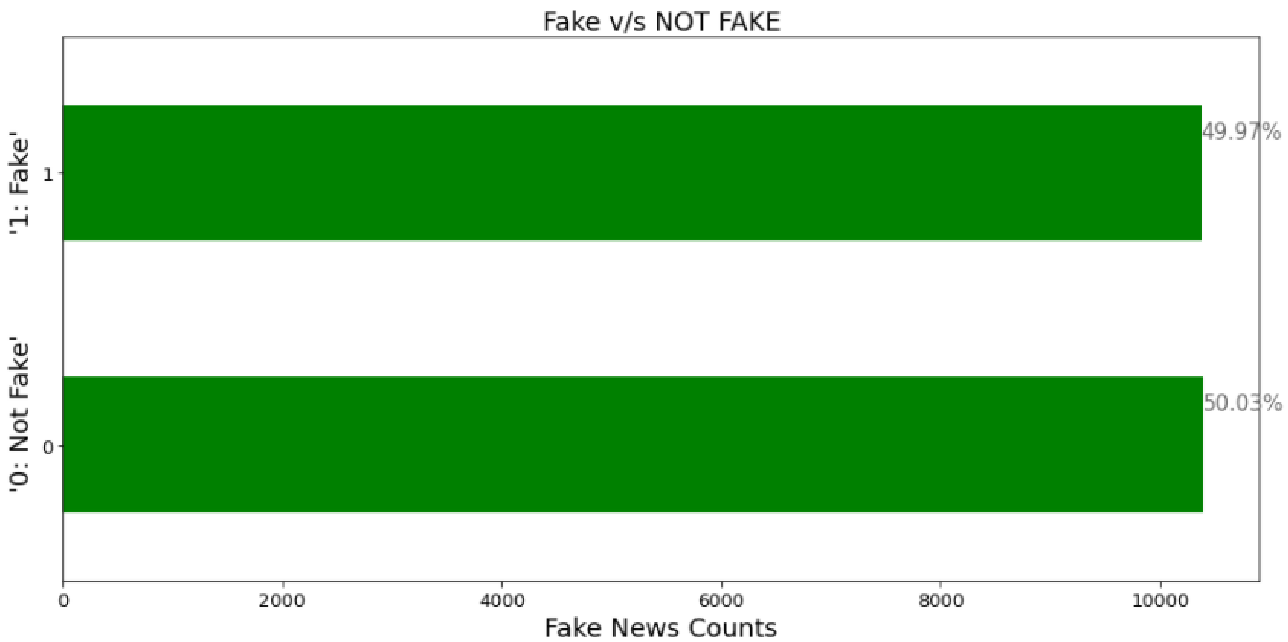
```python
# visualisation of Accuracy Score
sns.barplot(y=Model,x=Acc_score)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe195104750>
```



- Visualizations

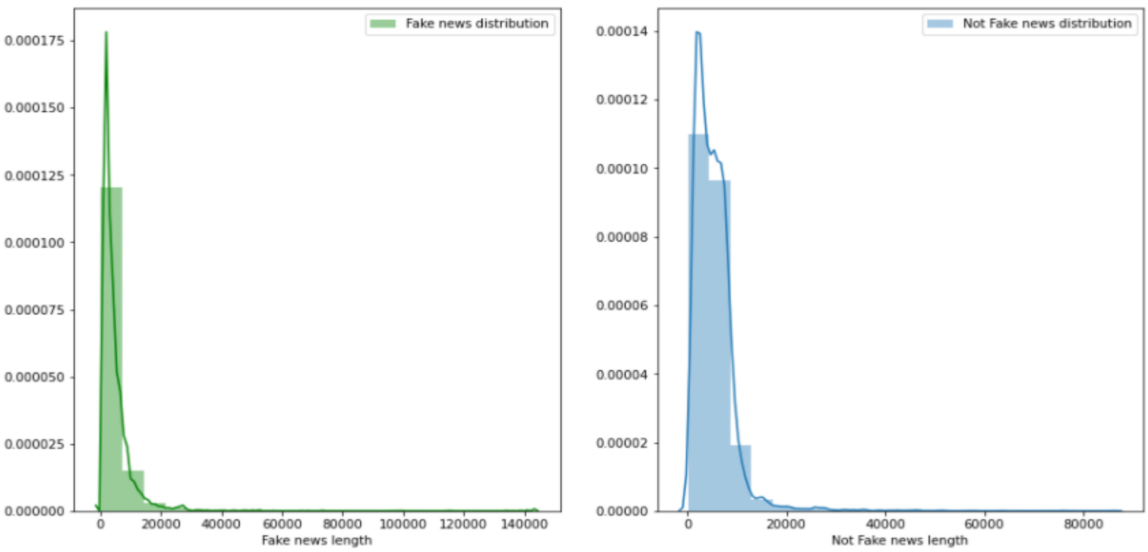
1. Count for Fake and Not Fake News



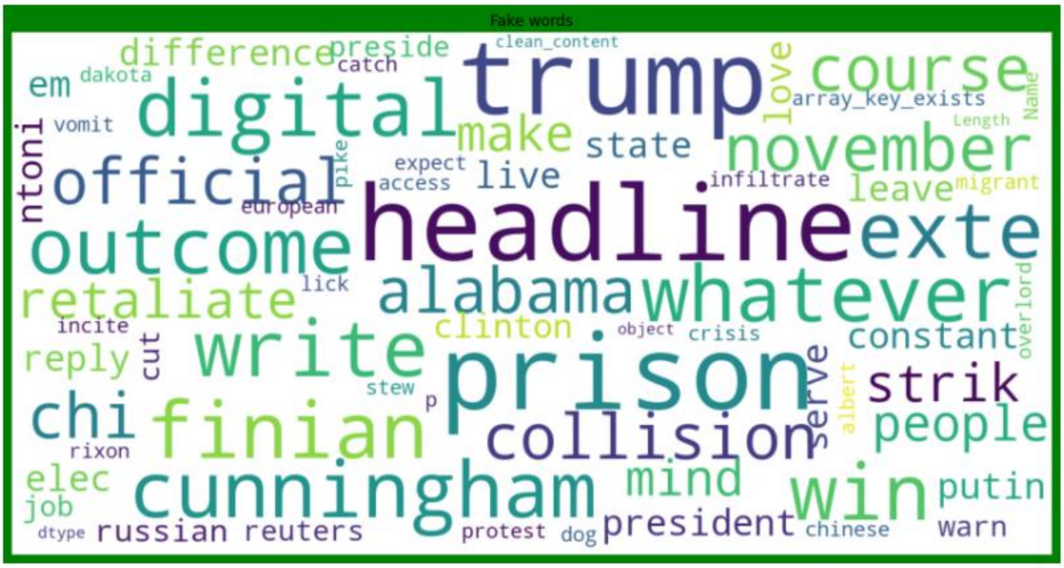
2. Fake Words before Cleaning

3. Not Fake Words before Cleaning

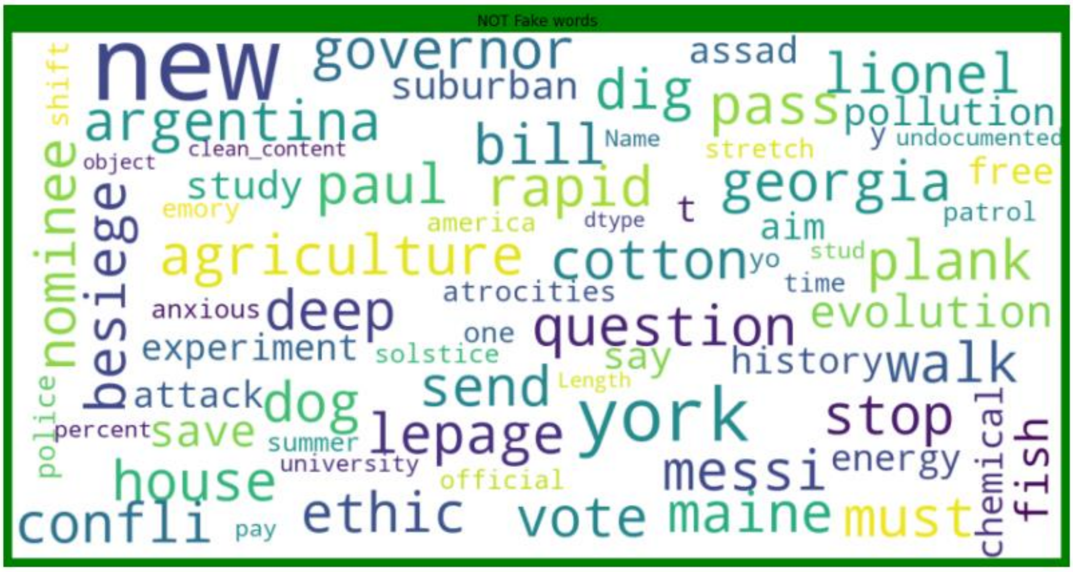
4. Length Distribution of Fake and Not Fake before cleaning



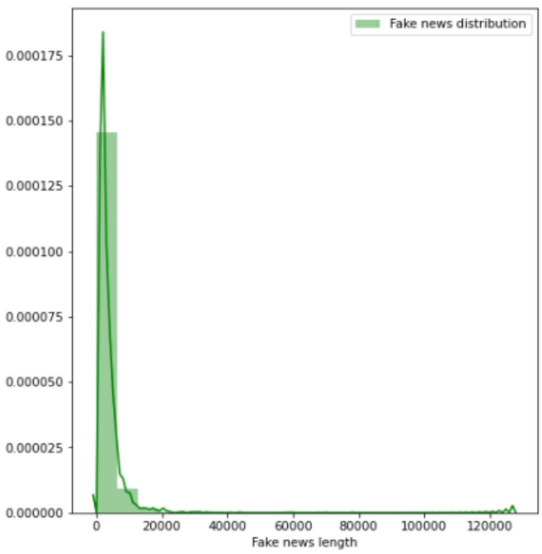
5. Fake Words after Cleaning

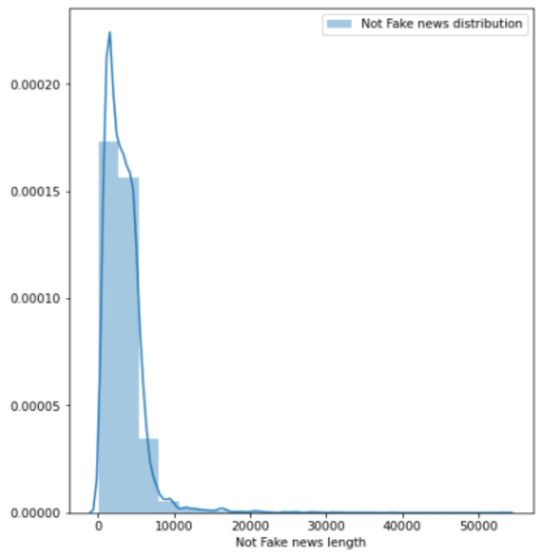


6. Not Fake Words after Cleaning



7. Length Distribution of Fake and Not Fake after cleaning





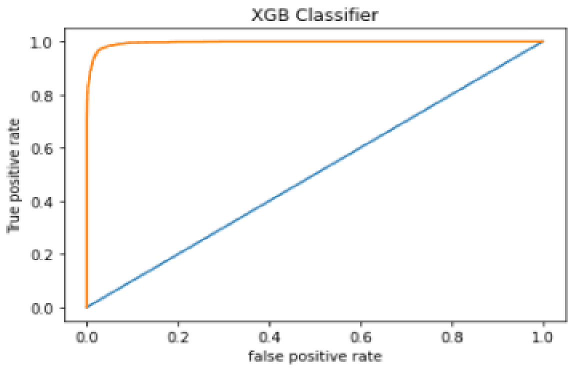
- Interpretation of Result

Final Model

XGB Classifier is giving an accuracy of 97% . So now I am making a final model using XGB Classifier.

```
# Using XGBClassifier for final model...
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=83,test_size=.30)
XG=XGBClassifier()
XG.fit(x_train,y_train)
XG.score(x_train,y_train)
XGpred=XG.predict(x_test)
print('Accuracy Score:',accuracy_score(y_test,XGpred))
print('Confusion Matrix:',confusion_matrix(y_test,XGpred))
print('Classification Report:','\n',classification_report(y_test,XGpred))
```

```
Accuracy Score: 0.9691764328142559
Confusion Matrix: [[2991  128]
 [  64 3046]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.96      0.97      3119
           1       0.96      0.98      0.97      3110

    accuracy                           0.97      6229
   macro avg       0.97      0.97      0.97      6229
weighted avg       0.97      0.97      0.97      6229
```




```
roc_auc_score =  0.9962263236732856
```

Prediction of y_test data

```
# Printing predicted values
test=pd.DataFrame(data=y_test,)
test['Predicted values']=XGpred
test
# On the lest side values are those which are taken by machine for test...
```

	0	Predicted values
0	1	1
1	0	0
2	1	1
3	0	0
4	0	0
...	...	...
6224	0	0
6225	1	1
6226	1	1
6227	1	1
6228	1	1

6229 rows × 2 columns

CONCLUSION

- **Key Findings and Conclusions of the Study**
  From the whole evaluation we can see that the maximum number of fake words used are Trump and Clinton and we can interrupt that it was due to election campaign held during US Presidential election and we know this adverse effect of the voters which were influenced by the fake news and the real had said trump and president and the fake news which was cleared by Trump's campaign but can hardly see clarity or real news from the side of Clinton and due to which the impact we already saw on election results and regarding the election advertisement and news Facebook's CEO Mark Zuckerberg also got extensively questioned by congress.

- **Learning Outcomes of the study with respect to Data science**
  So from the word frequency chart we can clearly see that most of the news is related to US Presidential election between Trump ad Clinton and by implementing passive aggressive algorithms we can see that we can achieved a good score as it calculates the error and updates its own learning rate which makes our model more reliable.

- **Limitations of this Work and Scope for Future Work**
  - I have also tried machine learning algorithm Gradient boosting Classifier which took enormous time more than a day (though not completed that's why I don't opt) to build the model.
  - Using hyperparameter tuning for XGB can increase our score.
  - Using Deep learning techniques may give some more good results.