FLIP ROBO

FAKE NEWS PREDICTION

Submitted by:

ArunPrasad.k

Table of content

Abstract*—The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in the human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article. In this work, we propose to use machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on 4 real world datasets. Experimental evaluation confirms the superior performance of our proposed ensemble learner approach in comparison to individual learners.

General Terms*: *Data Analytics, Exploratory Data Analytics, Machine Learning, Model Evaluation , Data Science.

Keywords- online comments, Malignantity, classifification models, TF-IDF technique, logistic regression.

1. INTRODUCTION

"Fake News" is a term used to represent fabricated news or propaganda comprising misinformation communicated through traditional media channels like print, and television as well as non-traditional media channels like social media. The general motive to spread such news is to mislead the readers, damage reputation of any entity, or to gain from sensationalism. It is seen as one of the greatest threats to democracy, free debate, and the Western order .Fake news is increasingly being shared via social media platforms like Twitter and Facebook. These platforms offer a setting for the general population to share their opinions and views in a raw and un-edited fashion. Some news articles hosted or shared on the social media platforms have more views compared to direct views from the media outlets' platform. Research that studied the velocity of fake news concluded that tweets containing false information reach people on Twitter six times faster than truthful tweets . The adverse effects of inaccurate news range from making people believe that Hillary Clinton had an alien baby, trying to convince readers that President Trump is trying to abolish first amendment to mob killings in India due to a false rumor propagated in WhatsApp.Technologies such as Artificial Intelligence (AI) and Natural Language Processing (NLP) tools offer great promise for researchers to build systems which could automatically detect fake news. However, detecting fake news is a challenging task to accomplish as it requires models to summarize the news and compare it to the actual news in order to classify it as fake. Moreover, the task of comparing proposed news with the original news itself is a daunting task as its highly subjective and opinionated. A different way to detect fake news is through stance detection which will be the focus of our study. Stance Detection is the process of automatically detecting the relationship between two pieces of text. In this study, we explore ways to predict the stance, given a news article and news headline pair. Depending on how similar the news article content and headlines are, the stances between them can be defined as 'agree', 'disagree', 'discuss' or 'unrelated'. We experimented with several traditional machine learning models to set a baseline and then compare results to the state-of-the-art deep networks to classify the stance between article body and headline. Through experimental procedures, we propose a model which can detect fake news by accurately predicting stance between the headline and the news article. We also studied how different hyper parameters affect the model performance and summarized the details for future work. Our model performs reasonably well when classifying between all the stances with some variations in accuracy for disagreed stances. Further we have discussed problems with defining and identifying fake news,

describe Fake News Challenge data set which we used to perform the experiment, and we discuss the previous work performed on similar problem. Then we provide a primer on various techniques used in our experiments such as natural language and deep learning. Next we explain the experimental design followed by our solution to solve the fake news detection problem in next Section .Then we present our results for different models and hyper parameter tuning for the models and we further conclude our findings in the Last.

2.PROBLEM STATEMENT

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 this project, we are given a dataset in the fake-news_data.zip folder. The folder contains a CSV files train_news.csv and you have to use the train_news.csv data to build a model to predict whether a news is fake or not fake. we have to try out different models on the dataset, evaluate their performance, and finally report the best model you got on the data and its performance.

3.DATASET

The data set contains the training set, which has approximately 20800 samples . The data samples contain 6 fields which includes **'Id', 'Headline', 'news', 'Unnamed:0', 'Written_by', 'label'.**

4. DATA OVERVIEW

There are 6 columns in the dataset provided to us. 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).

5. APPROACH

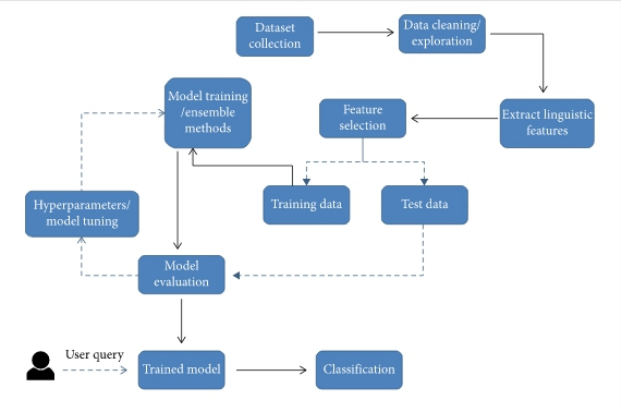


Fig. 1 System Design of the Study

The steps for document classification are:

i. Read Document

ii. Tokenization, where every word in the document is split into tokens

iii. Removing stop words, punctuation, or unwanted tokens

iv. Lemmatization/Stemming, Shorten words to their root stems, e.g. to take walk, walked, walking, walks as Walk etc

v. Feature vector representation

vi. Feature extraction

vii. Learning algorithm

6.IMPLEMENTATION

6.1. Importing Train Dataset:

Reading the data to plot the graphs:

```
import io
df = pd.read_csv(io.BytesIO(uploaded['train_news.csv']))
df.head(10)
```

	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
5	5	3196	Whatever the Outcome on November 8th the US Wi...	NaN	Taming the corporate media beast Whatever the ...	1
6	6	5134	Rapid Evolution Saved This Fish From Pollution...	JoAnna Klein	The State of New Jersey says you can't eat the...	0
7	7	1504	Alabama Prison Officials Retaliate Against Pri...	Brian Sonenstein	Advocates say prison officials at the Kilby Co...	1
8	8	13559	NaN	steventexas	People have made up their minds on president \...	1
9	9	4203	Can We Live in a Constant State of Love?	Gillian	Leave a reply \nToni Emerson – When we fall in...	1

6.2 Cleaning NAN values:

Before applying any type of data analytics on the data set, the data should be first cleaned.

Lets check for any missing values.

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

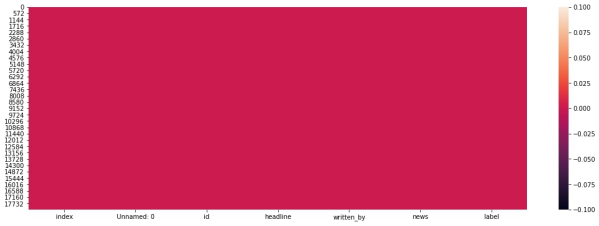
```
Unnamed: 0        0
id                0
headline        558
written_by     1957
news             39
label             0
dtype: int64
```

- Here we see Missing values in column **headline,written_by,news**
- So droping of missing values is required

So removing all NaN values using dropna() method.

```
#Removing NaN values
df.dropna(axis="rows",inplace=True)
df
```

```
#Graphical Representation
#Heatmap
plt.figure(figsize=(18,6))
sns.heatmap(message.isnull())
```



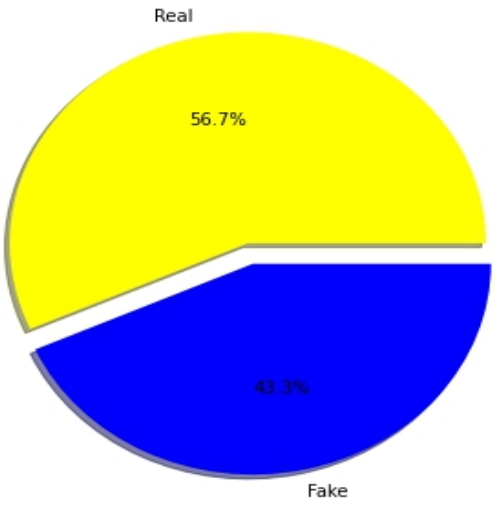
7. EXPLORATORY DATA ANALYSIS

We are going to perform exploratory data analysis for our problem in the first stage. In exploratory data analysis data set is explored to figure out the features which would influence the Fake news prediction. The data is deeply analyzed by finding a relationship between each attribute and our target label.

7.1 Uni-variate Analysis

i) Target - "label"

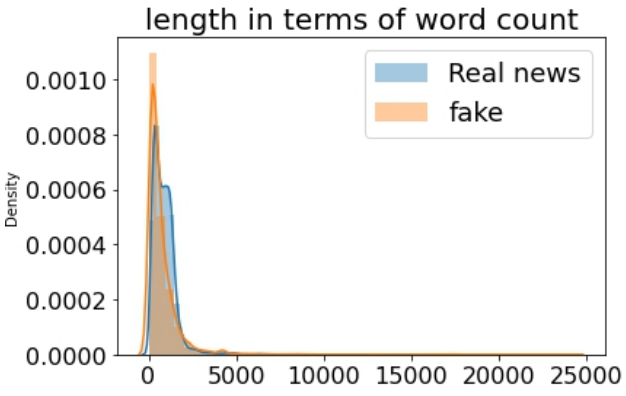
```
#Visuvalization
plt.figure(figsize=(8,6))
sizes = [10361,7924]
labels = ['Real','Fake']
colors = ['yellow', 'blue']
explode = [0.1,0]
plt.pie(sizes,labels= labels,colors= colors,explode = explode, autopct = '%1.1f%%',shadow = True)
plt.show()
```



ii) Exploration on Content

```
real = df[df.label == 0]['content'].apply(lambda x: len(x.split()))
fake = df[df.label == 1]['content'].apply(lambda x: len(x.split()))
```

```
#displacement graph
sns.distplot(real, label='Real news')
sns.distplot(fake, label='fake')
plt.legend(fontsize=18)
plt.xlabel(None)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.title("length in terms of word count", fontsize=20)
```



8.FEATURE ENGINEERING

8.1 CLEANING THE TEXT

Since, the content in the dataset were collected from the internet they may contain various elements in them.

First we will be Removing Punctuation and other special characters.We then convert each comment into lower case and then split it into individual words. There were some words in the dataset which had length > 100, since there are no words in the English language whose length > 100, we remove such words.

First, we tried building the features removing stop words and then trained some models thinking that it may help the model in learning the semantics of fake words, but we found out that the model learns better if there are stop words in the comment.

Possible reason is, generally a fake news is used from the internet, seeing the data we found out that those persons are generally referred by pronouns, which are nothing but stop words.

8.2 LEMMATIZERS

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove endings only and to return the base or dictionary form of a word, which is known as the lemma.

We used **WordNet lemmatizer**. For grammatical reasons, documents are going to use different forms of a word, such as organizes, organize and organizing. But they all represent the same semantics. So, Lemmatizer for those three words gives a single word, which helps algorithm learn better.

```
wordnet = WordNetLemmatizer()
def lemmatize(content):
  #removing Puntuations and symbols
  words = re.sub('[^a-zA-Z]', ' ',content)
  #Replacing all characters to lower case
  words = words.lower()
  #Spilitting every words
  words = words.split()
  #Removing stopwords and Lemmatizing
  words = [wordnet.lemmatize(word) for word in words if not word in stopwords.words('english')]
  words = ' '.join(words)
  return words
```

```
df['content'] = df['content'].apply(lemmatize)
```

8.3 VECTORIZATION

Python's scikit-learn deals with numeric data only. To convert the text data into numerical form, tf-idf vectorizer is used. TF-IDF vectorizer converts a collection of raw documents to a matrix of Tf-idf features.

We set the predictor variable on the dataset with tf-idf vectorizer, in two different ways. First, by setting the parameter analyzer as 'word'(select words) and the second by setting it to 'char'(select characters). Using 'char' was important because the data had many 'foreign languages' and they were difficult to deal with by considering only the 'word' analyzer.

We set the parameter n-gram range (an n-gram is a continuous sequence of n-items from a given sample of text or speech). After trying various values, we set the n-gram as (1, 3) for 'word' analyzer. We also set the max_features as 30000 for both word and character analyzer after many trails.

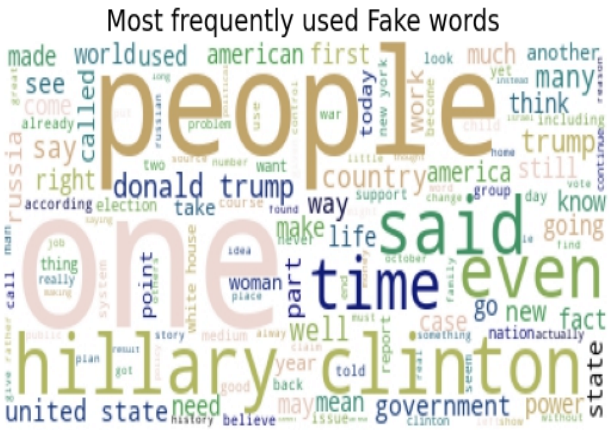
We then combined the word and character features and transformed the dataset into two sparse matrixes for train and test sets, respectively using tf-idf vectorizer.

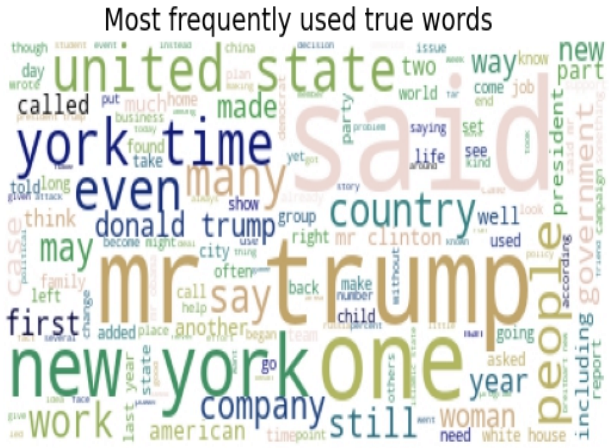
```
# converting the textual data to numerical data
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
x=vectorizer.fit_transform(x)
```

8.4 Word cloud

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

Here we will be showing Most frequently used fake and not a fake words in a visualization chart.





9. DATA TRAINING

Training of the processed dataset is done by problem as a multi-label classification problem. Multi-label classification problem is transformed single-class classifier problem. This is known as problem transformation. This is achieved using the binary relevance method. The pre-processed dataset is trained using

1. **LogisticRegression**
2. **KNeighborsClassifier**
3. **MultinomialNB**
4. **BernoulliNB**
5. **LinearSVC**

```
model=[lr,knn,bnb,mnb,svm]
for m in model:
    m.fit(x_train,y_train)
    pred_train=m.predict(x_train)
    pred_test=m.predict(x_test)
    plot_confusion_matrix(m, x_test, y_test)
    print("the score of ",m,"is")
    print("training accuracy score  :",accuracy_score(y_train,pred_train)*100)
    print("testing accuracy score  :",accuracy_score(y_test,pred_test)*100)
    plt.show()
    print("**********************************************************")
    print("\n\n")
```

```
model=[lr,knn,bnb,mnb,svm]
for m in model:
    m.fit(x_train,y_train)
    pred_train=m.predict(x_train)
    pred_test=m.predict(x_test)
    print("Report of ",m, "is")
    print("F1 score \n",f1_score(y_test,pred_test)*100)
    print("classification_report  \n",classification_report(y_test,pred_test))
```

MODEL	TRAINING ACCURACY	TESTING ACCURACY	F1 ACCURACY
Logistic	97.94	95.70	95.02
Kneighbour	54.29	51.46	63.95
Multinobial NB	82.46	79.46	69.10
BernoulliNB	84.01	79.71	79.30
LinearSCV	99.94	97.56	97.20

10.PREDICTION

Since we have evaluated all models by using confusion matrix we will select the best by using model which has highest accuracy.Here we can choose **LinearSVC** model to predict the Fake news.

10.1. Hyper Tuning using GridSearchCV

Hyperparameters are crucial as they control the overall behavior of a machine learning model. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results. Here we first find the best parameters for the Random Forest Model and indulge it to the model to improve our predicting accuracy. There are several ways for finding the parameters, here we use the most powerful and most commonly used method named GridSearchCV.

```
#Hyper parameter Tuning
#Linear Support Vector Classification
#using GridSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn.svm import LinearSVC
parameters={"C" :[0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0],
            "random_state": list(range(0,10)),
            "max_iter": [2,3,4,5,6,7,8],
            "penalty":["l1","l2"],
            "loss":["hinge", "squared_hinge"]}
svm = LinearSVC()
clf=GridSearchCV(svm,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)
```

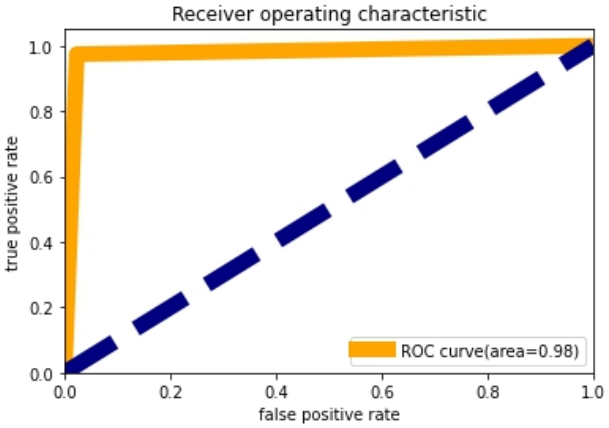
```
'C': 1.0, 'loss': 'squared_hinge',

'max_iter': 8, 'penalty': 'l2'

'random_state': 6
```

10.2 Further Evaluation

i) Cross Validation

The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

ii) AUC-ROC CURVE



11. FINAL PREDICTED OUTPUT

```
import numpy as np
a=np.array(y_test)
predicted=np.array(svm.predict(x_test))
df_con=pd.DataFrame({"original":a,"Predicted":predicted})
df_con.head(10)
```

	original	Predicted
0	1	1
1	1	1
2	0	0
3	0	0
4	0	0
5	1	1
6	0	0
7	0	0
8	0	0
9	0	0

12.CONCLUSION

The task of classifying news manually requires in-depth knowledge of the domain and expertise to identify anomalies in the text. In this research, we discussed the problem of classifying fake news articles using machine learning models and ensemble techniques. The data we used in our work is collected from the World Wide Web and contains news articles from various domains to cover most of the news rather than specifically classifying political news.

The primary aim of the research is to identify patterns in text that differentiate fake articles from true news. We extracted different textual features from the articles using an LIWC tool and used the feature set as an input to the models. The learning models were trained and parameter-tuned to obtain optimal accuracy. Some models have achieved comparatively higher accuracy than others. We used multiple performance metrics to compare the results for each algorithm. The ensemble learners have shown an overall better score on all performance metrics as compared to the individual learners.

Fake news detection has many open issues that require attention of researchers. For instance, in order to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Graph theory and machine learning techniques can be employed to identify the key sources involved in spread of fake news. Likewise, real time fake news identification in videos can be another possible future direction.

REFERENCE

I. A. Douglas, “News consumption and the new electronic media,” The International Journal of Press/Politics, vol. 11, no. 1, pp. 29–52, 2006.View at: Publisher Site | Google Scholar

II. J. Wong, “Almost all the traffic to fake news sites is from facebook, new data show,” 2016.View at: Google Scholar

III. D. M. J. Lazer, M. A. Baum, Y. Benkler et al., “The science of fake news,” Science, vol. 359, no. 6380, pp. 1094–1096, 2018.View at: Publisher Site | Google Scholar

IV. S. A. García, G. G. García, M. S. Prieto, A. J. M. Guerrero, and C. R. Jiménez, “The impact of term fake news on the scientific community scientific performance and mapping in web of science,” Social Sciences, vol. 9, no. 5, 2020.View at: Google Scholar

V. A. D. Holan, 2016 Lie of the Year: Fake News, Politifact, Washington, DC, USA, 2016.

VI. S. Kogan, T. J. Moskowitz, and M. Niessner, “Fake News: Evidence from Financial Markets,” 2019, https://ssrn.com/abstract=3237763.View at: Google Scholar

VII. A. Robb, “Anatomy of a fake news scandal,” Rolling Stone, vol. 1301, pp. 28–33, 2017.View at: Google Scholar

VIII. J. Soll, “The long and brutal history of fake news,” Politico Magazine, vol. 18, no. 12, 2016.View at: Google Scholar

IX. J. Hua and R. Shaw, “Corona virus (covid-19) “infodemic” and emerging issues through a data lens: the case of China,” International Journal of Environmental Research and Public Health, vol. 17, no. 7, p. 2309, 2020.View at: Publisher Site | Google Scholar

X. N. K. Conroy, V. L. Rubin, and Y. Chen, “Automatic deception detection: methods for finding fake news,” Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1–4, 2015.View at: Publisher Site | Google Scholar