VIT-AP UNIVERSITY

PROJECT REPORT

CSE4027 |SLOT:E

Submitted to: PROF.GOPIKRISHNAN

FAKE NEWS DETECTION

TEAM MEMBERS:

TEAM LEAD & DATA ANALYST: K.Shiva Kalyan Kumar

(19BCI7076)

TEAM LEAD& DOCUMENTATION: P.Bindu Madhavi

(19BCE7124)

DATA SCIENTIST: G.Bharath Sai

(19BCE7556)

INDEX:

DECLARATION:

We, K. Shiva Kalyan Kumar(19BCI7076) P. Bindu Madhavi (19BCE7124), and G.Bharath Sai (19BCE7556)of 3^{rd} year B.Tech., in the department of Computer Science and Engineering from Vellore Institute of Technology, Amaravathi, hereby declare that the project work entitled FAKE NEWS DETECTION using R programming is carried out by us and worked under Prof.Gopikrishnan sir. We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

ABSTRACT:

In our modern era where the internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in the use of social media platforms like Facebook, Twitter, Instagram, etc. news spread rapidly among millions of users within a very short period. The spread of fake news has far-reaching consequences like the creation of biased opinions to sway election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits.

In this project, we aim to provide the user with the ability to classify the news as fake or real.

PROBLEM STATEMENT:

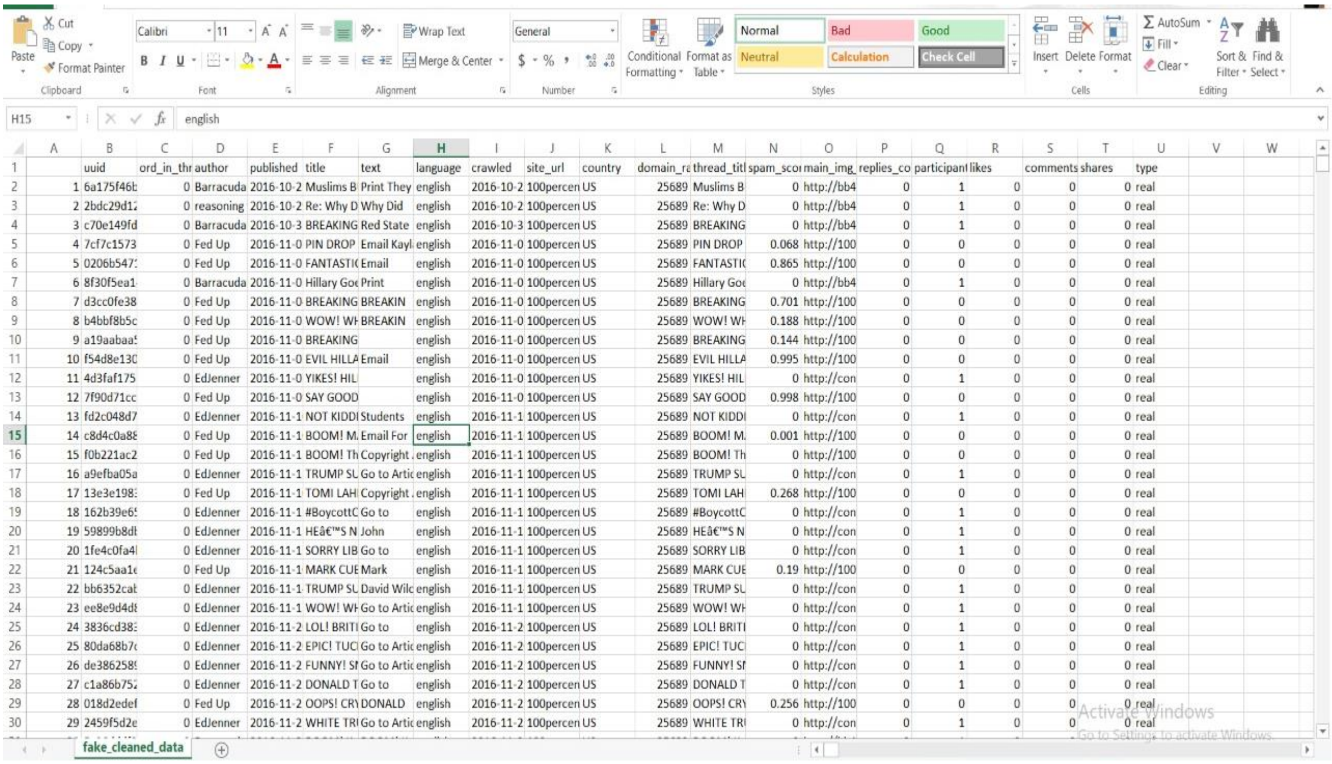
News consumption is a double-edged sword. On the one hand, its low cost, easy access, and rapid dissemination of information lead people to seek out and consume news. It enables the widespread of "fake news", i.e., low-quality news with intentionally false information. The extensive spread of fake news has the potential for extremely negative impacts on individuals and society.

Therefore, fake news detection has recently become emerging research that is attracting tremendous attention. First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content. To develop a FAKE NEWS DETECTION system using R language and machine learning model decision tree and its accuracy will be tested.

ABOUT DATASET:

This dataset includes recent stories covering fake news in news. This is a sensitive, nuanced topic. From defining fake, biased, and misleading news in the first place to deciding how to take action, there's a lot of information to consider beyond what can be neatly arranged in a CSV file.

The dataset contains text and metadata from 244 websites and represents 12,999 posts in total from the past 30 days. The data was pulled using the webhose.io API; because it's coming from their crawler, not all websites identified by the BS Detector are present in this dataset. . There are no genuine, reliable, or trustworthy news sources represented in this dataset.



	uuid	ord_in_thr	author	published	title	text	language	crawled	site_url	country	domain_ra	thread_titl	spam_scor	main_img_	replies_co	participant	likes	comments	shares	type
1	6a175f46b	0	Barracuda	2016-10-2	Muslims B	Print They	english	2016-10-2	100percen	US	25689	Muslims B	0	http://bb4	0	1	0	0	0	real
2	2bdc29d12	0	reasoning	2016-10-2	Re: Why D	Why Did	english	2016-10-2	100percen	US	25689	Re: Why D	0	http://bb4	0	1	0	0	0	real
3	c70e149fd	0	Barracuda	2016-10-3	BREAKING	Red State	english	2016-10-3	100percen	US	25689	BREAKING	0	http://bb4	0	1	0	0	0	real
4	7cf7c1573	0	Fed Up	2016-11-0	PIN DROP	Email Kayl	english	2016-11-0	100percen	US	25689	PIN DROP	0.068	http://100	0	0	0	0	0	real
5	0206b547	0	Fed Up	2016-11-0	FANTASTIC	Email	english	2016-11-0	100percen	US	25689	FANTASTIC	0.865	http://100	0	0	0	0	0	real
6	8f30f5ea1	0	Barracuda	2016-11-0	Hillary Goe	Print	english	2016-11-0	100percen	US	25689	Hillary Goe	0	http://bb4	0	1	0	0	0	real
7	d3cc0fe38	0	Fed Up	2016-11-0	BREAKING	BREAKIN	english	2016-11-0	100percen	US	25689	BREAKING	0.701	http://100	0	0	0	0	0	real
8	b4bbf8b5c	0	Fed Up	2016-11-0	WOW! WH	BREAKIN	english	2016-11-0	100percen	US	25689	WOW! WH	0.188	http://100	0	0	0	0	0	real
9	a19aabaa	0	Fed Up	2016-11-0	BREAKING		english	2016-11-0	100percen	US	25689	BREAKING	0.144	http://100	0	0	0	0	0	real
10	f54d8e130	0	Fed Up	2016-11-0	EVIL HILLA	Email	english	2016-11-0	100percen	US	25689	EVIL HILLA	0.995	http://100	0	0	0	0	0	real
11	4d3faf175	0	EdJenner	2016-11-0	YIKES! HIL		english	2016-11-0	100percen	US	25689	YIKES! HIL	0	http://con	0	1	0	0	0	real
12	7f90d71cc	0	Fed Up	2016-11-0	SAY GOOD		english	2016-11-0	100percen	US	25689	SAY GOOD	0.998	http://100	0	0	0	0	0	real
13	fd2c048d7	0	EdJenner	2016-11-1	NOT KIDDI	Students	english	2016-11-1	100percen	US	25689	NOT KIDDI	0	http://con	0	1	0	0	0	real
14	c8d4c0a88	0	Fed Up	2016-11-1	BOOM! M.	Email For	english	2016-11-1	100percen	US	25689	BOOM! M.	0.001	http://100	0	0	0	0	0	real
15	f0b221ac2	0	Fed Up	2016-11-1	BOOM! Th	Copyright .	english	2016-11-1	100percen	US	25689	BOOM! Th	0	http://100	0	0	0	0	0	real
16	a9efba05a	0	EdJenner	2016-11-1	TRUMP SU	Go to Artic	english	2016-11-1	100percen	US	25689	TRUMP SU	0	http://con	0	1	0	0	0	real
17	13e3e198	0	Fed Up	2016-11-1	TOMI LAH	Copyright .	english	2016-11-1	100percen	US	25689	TOMI LAH	0.268	http://100	0	0	0	0	0	real
18	162b39e65	0	EdJenner	2016-11-1	#BoycottC	Go to	english	2016-11-1	100percen	US	25689	#BoycottC	0	http://con	0	1	0	0	0	real
19	59899b8db	0	EdJenner	2016-11-1	HEâ€™S N	John	english	2016-11-1	100percen	US	25689	HEâ€™S N	0	http://con	0	1	0	0	0	real
20	1fe4c0fa4	0	EdJenner	2016-11-1	SORRY LIB	Go to	english	2016-11-1	100percen	US	25689	SORRY LIB	0	http://con	0	1	0	0	0	real
21	124c5aa1	0	Fed Up	2016-11-1	MARK CUE	Mark	english	2016-11-1	100percen	US	25689	MARK CUE	0.19	http://100	0	0	0	0	0	real
22	bb6352cab	0	EdJenner	2016-11-1	TRUMP SU	David Wilc	english	2016-11-1	100percen	US	25689	TRUMP SU	0	http://con	0	1	0	0	0	real
23	ee8e9d4d8	0	EdJenner	2016-11-1	WOW! WH	Go to Artic	english	2016-11-1	100percen	US	25689	WOW! WH	0	http://con	0	1	0	0	0	real
24	3836cd38	0	EdJenner	2016-11-2	LOL! BRITI	Go to	english	2016-11-2	100percen	US	25689	LOL! BRITI	0	http://con	0	1	0	0	0	real
25	80da68b7	0	EdJenner	2016-11-2	EPIC! TUC	Go to Artic	english	2016-11-2	100percen	US	25689	EPIC! TUC	0	http://con	0	1	0	0	0	real
26	de386258	0	EdJenner	2016-11-2	FUNNY! SM	Go to Artic	english	2016-11-2	100percen	US	25689	FUNNY! SM	0	http://con	0	1	0	0	0	real
27	c1a86b75	0	EdJenner	2016-11-2	DONALD T	Go to	english	2016-11-2	100percen	US	25689	DONALD T	0	http://con	0	1	0	0	0	real
28	018d2edef	0	Fed Up	2016-11-2	OOPS! CRY	DONALD	english	2016-11-2	100percen	US	25689	OOPS! CRY	0.256	http://100	0	0	0	0	0	real
29	2459f5d2e	0	EdJenner	2016-11-2	WHITE TRI	Go to Artic	english	2016-11-2	100percen	US	25689	WHITE TRI	0	http://con	0	1	0	0	0	real

IMPLEMENTATION :

WORK DONE BY DATA SCIENTIST:

#Loading the libraries

library("tidyverse")

library("tidytext") # tidy implimentation of NLP methods

library("syuzhet")

```
> library("tidyverse")
> library("tidytext") # tidy implimentation of NLP methods
> library("syuzhet")
```

news<-read.csv("fake.csv")

#finding dimensions

```
> news<-read.csv("D:/fake.csv")
> dim(news)
[1] 12999    20
> sum(is.na(news))
[1] 4223
> #doing for each row
> sum(is.na(news$uuid))
[1] 0
> sum(is.na(news$ord_in_thread))
[1] 0
> sum(is.na(news$author))
[1] 0
> sum(is.na(news$published))
[1] 0
> sum(is.na(news$title))
[1] 0
> sum(is.na(news$text))
[1] 0
> sum(is.na(news$language))
[1] 0
> sum(is.na(news$crawled))
[1] 0
> sum(is.na(news$site_url))
[1] 0
> sum(is.na(news$country))
[1] 0
> sum(is.na(news$domain_rank))
[1] 4223
```

#Here we have NA values in the domain_rank column
#so set default value -15

```
> news$domain_rank[is.na(news$domain_rank)] <- 15
> sum(is.na(news))
[1] 0
> write.csv(news,file = "fake_cleaned_data.csv")
> sum(is.na(news$author))
[1] 0
```

#bs and conspiracy news are also fake

```
> news$type<-gsub("bs","fake",news$type)
> news$type<-gsub("conspiracy","fake",news$type)
```

#while others are real

news$type<-gsub("bias","real",news$type)

news$type<-gsub("satire","real",news$type)

news$type<-gsub("hate","real",news$type)

news$type<-gsub("junksci","real",news$type)

news$type<-gsub("state","real",news$type)

```
> #while others are real
> news$type<-gsub("bias","real",news$type)
> news$type<-gsub("satire","real",news$type)
> news$type<-gsub("hate","real",news$type)
> news$type<-gsub("junksci","real",news$type)
> news$type<-gsub("state","real",news$type)
```

#Count of type of news that how many are fake and real

```
> news %>% group_by(type) %>% summarise(count=n())
# A tibble: 2 x 2
  type  count
  <chr> <int>
1 fake  11941
2 real   1058
```

#apply function for finding question marks and exclamations

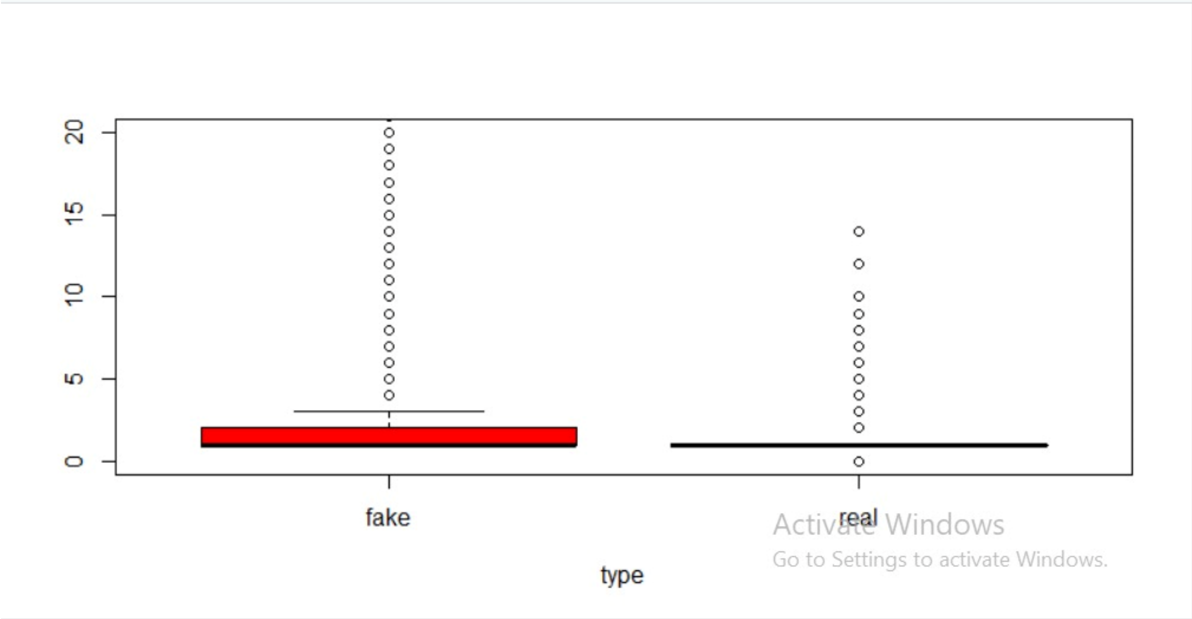
```
> news$exc <- sapply(news$text, function(x) length(unlist(strsplit(as.character(x),
 "\\!+")))) #count exclamation
> news$que <- sapply(news$text, function(x) length(unlist(strsplit(as.character(x),
 "\\?+")))) #count question marks
> ##Count of exclamations in fake and real news
> news %>% group_by(type) %>% summarise(exclamations=sum(exc))
# A tibble: 2 x 2
  type  exclamations
  <chr>        <int>
1 fake         20895
2 real          1678
```

#Count of question marks in fake and real news

```
> #Count of question marks in fake and real news
> news %>% group_by(type) %>% summarise(QuestionMarks=sum(que))
# A tibble: 2 x 2
  type  QuestionMarks
  <chr>         <int>
1 fake          31663
2 real           2241
```

#boxplot for exclamations in fake and real news

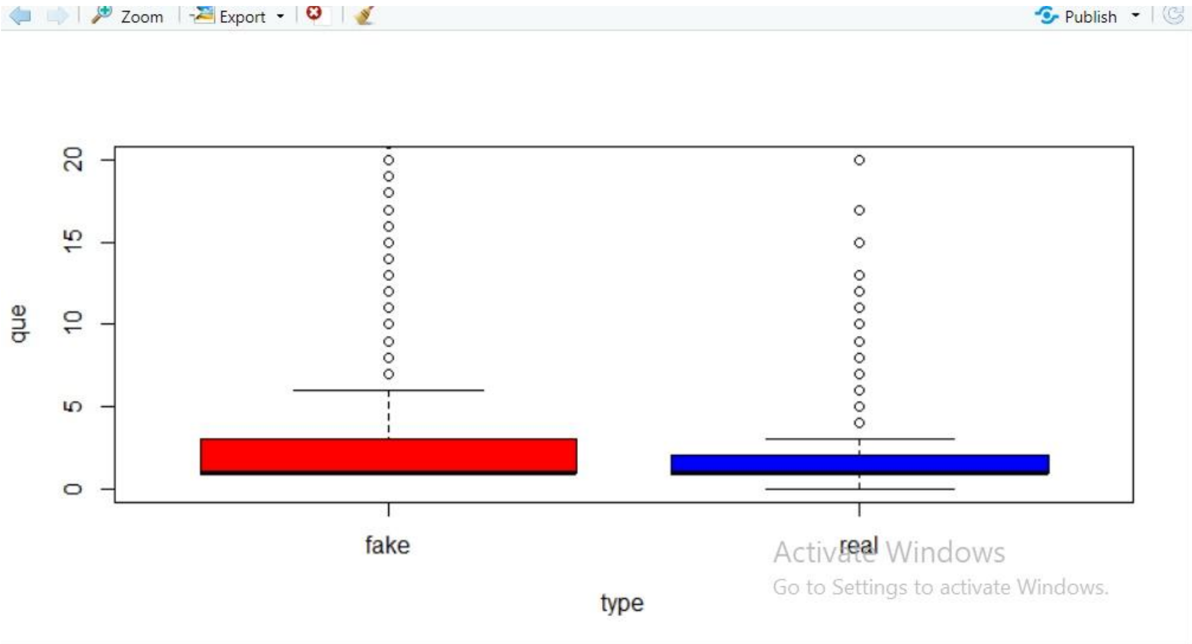
```
> boxplot(exc ~ type,news,ylim=c(0,20),ylab="",col=c("red","blue"))
```



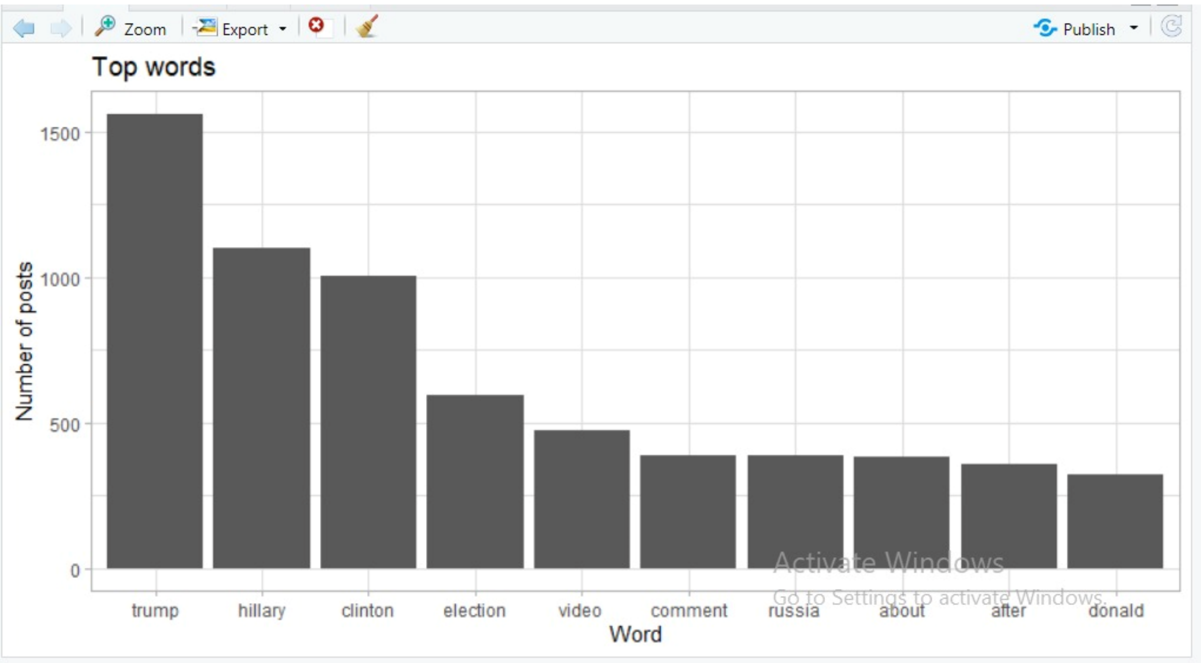
#we can observe that fake news have more exclamations than real news

#boxplot for question marks in fake and real news

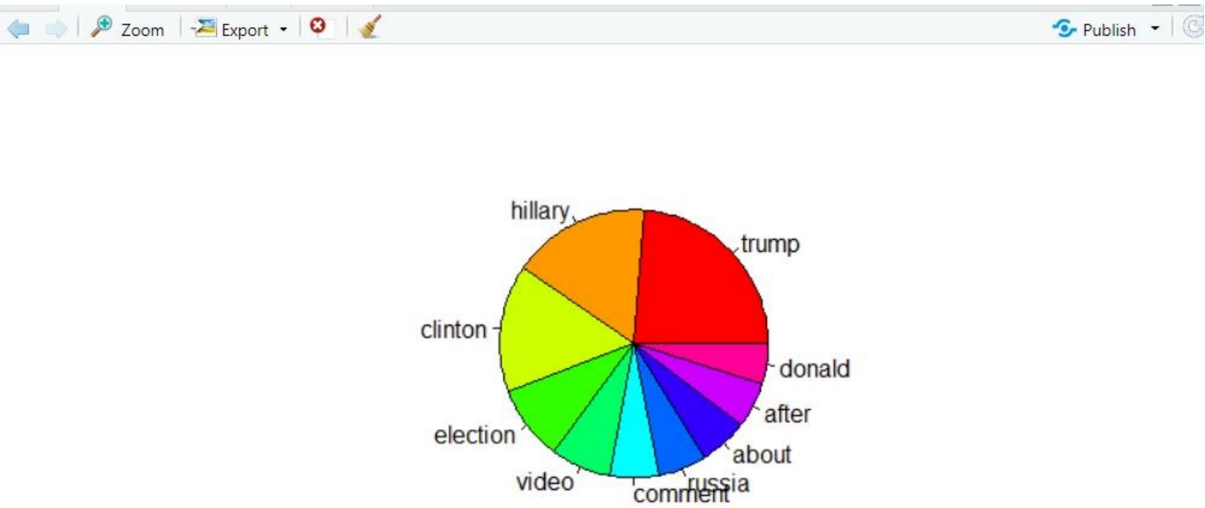
```
> boxplot(que ~ type,news,ylim=c(0,20),col=c("red","blue"))
```



```
> mytext = data_frame(text = news$title) %>%
+   unnest_tokens(word, text) %>%
+   group_by(word) %>%
+   count(word, sort = TRUE) %>% mutate(len=nchar(word)) %>% filter(len>4)
> pl = ggplot(head(mytext,10), aes(x=reorder(word, -n),y=n)) +
+   geom_col() +
+   theme_light() +
+   ylab("Number of posts") +
+   xlab("Word") +
+   ggtitle("Top words")
> pl
```




```
> pie(head(mytext$n,10),labels = mytext$word,col = rainbow(10))
```



#we can observe that fake news have more question marks than real

terms<- function(fake, text_column, group_column){

group_column <- enquo(group_column)

text_column <- enquo(text_column)

get the count of each word in each review

words <- news %>%

unnest_tokens(word, !!text_column) %>%

count(!!group_column, word) %>%

ungroup()

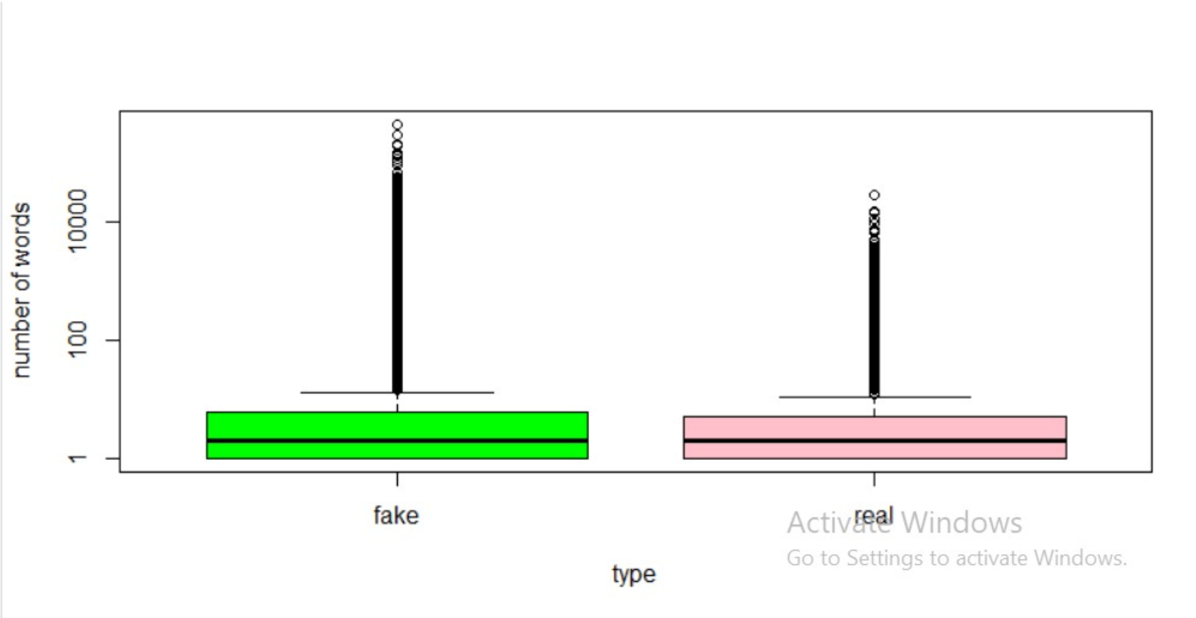
```
+   text_column <- enquo(text_column)
+
+   # get the count of each word in each review
+   words <- news %>%
+     unnest_tokens(word, !!text_column) %>%
+     count(!!group_column, word) %>%
+     ungroup()
+
+   # get the number of words per text
+   #total_words <- words %>%
+   #group_by(!!group_column) %>%
+   #summarize(total = sum(n))
+
+   # combine the two dataframes we just made
+
+   return (words)
+ }
```

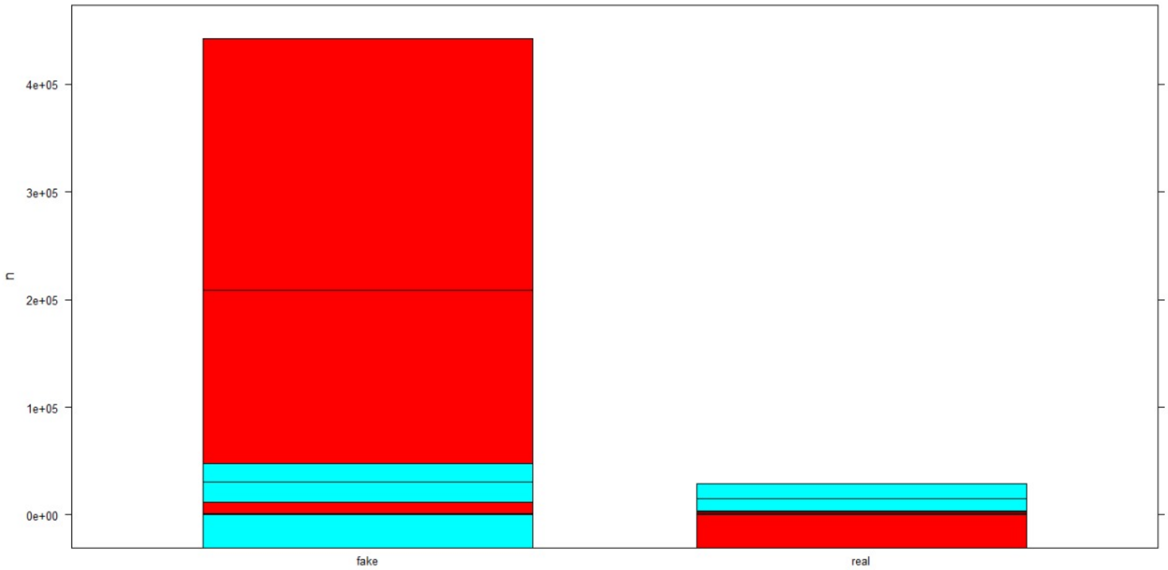
#store all words per text in a different data frame

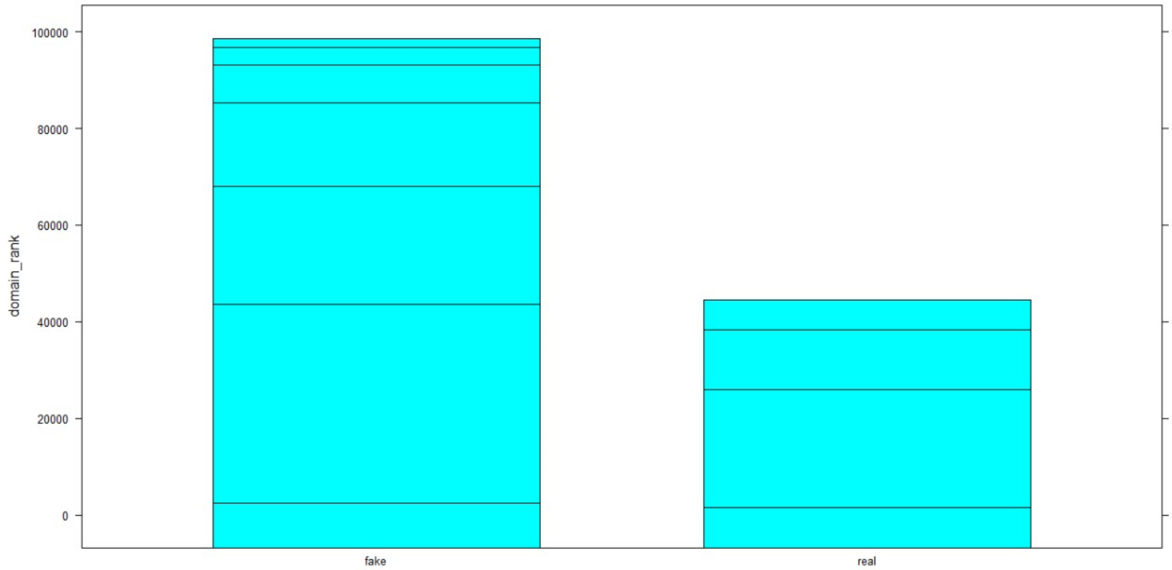
```
> df<-terms(news,text,type)
```

#create boxplot for number of words of each type

```
> boxplot(n ~ type,df,log="y",xlab="type",ylab="number of words",col=c("green","pin
k"))
```


number of words
1
100
10000
fake
real
type
Activate Windows
Go to Settings to activate Windows.


n
0e+00
1e+05
2e+05
3e+05
4e+05
fake
real



WORK DONE BY DATA ANALYST:

```
> library("tidyverse")
> library("tidytext") # tidy implimentation of NLP methods
> library("syuzhet")
> library("party")
> library("rpart")
> library("rpart.plot")
> news = read.csv("D:/fake_cleaned_data_1.csv")
> news1 = news[sample(1:nrow(news)), ]
> news1$type = factor(news1$type)
> tts = sample.split(news1,SplitRatio = 0.8)
> train = subset(news1,tts == T)
> test = subset(news1,tts == F)
> #Finding Sentiment of each news
> sentiment<-get_nrc_sentiment(train$text)
> head(sentiment)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     9            9       8   12   4       5        4    17       25       15
2     9           10       4   15   4       7        3     9       15       10
3     0            3       0    1   3       0        0     9        2       13
4    23           29      12   21  24      10       18    54       37       82
5     0            3       0    0   5       0        2     7        1       13
6     3            3       1    3   1       1        3     4        4        5
> sentiment1<-get_nrc_sentiment(test$text)
> head(sentiment1)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     5            2       2    7   1       6        1     4        6        6
2    49           41      17   47  21      30       13    65       85       99
3     2            2       3    4   1       1        2     5        7       11
4     9            5       6    8   5       7        8    10       13       15
5     3            5       1    3   6       2        3    11        5       16
6    14            6       7   15   6      10        5    10       16       16
> #taking only last two columns negative and positive for the analysis
> df1<-sentiment[c(9,10)]
> df2 = sentiment1[c(9,10)]
> #function for normalization
> normalize <- function(x) {
+   return ((x - min(x)) / (max(x) - min(x)))
+ }
```

```
> #normalize negative and positive column for better analysis means the values will lie between
 0 and 1
> df1$negative<-normalize(df1$negative)
> df1$positive<-normalize(df1$positive)
> #Combine this with the news dataset
> train<-cbind(train,df1)
> #finding standard deviations and median of negative and positive columns for each type of news

> neg_sd<-train %>% group_by(type) %>% summarise(neg_sd=sd(negative))
> pos_sd<-train %>% group_by(type) %>% summarise(pos_sd=sd(positive))
> neg_med<-train %>% group_by(type) %>% summarise(neg_med=median(negative))
> pos_med<-train %>% group_by(type) %>% summarise(pos_med=median(positive))
> #create dataframes for negative and positive standard deviations and median
> dfr2<-data.frame(neg_sd)
> dfr1<-data.frame(pos_sd)
> dfr3<-data.frame(neg_med)
> dfr4<-data.frame(pos_med)
```

#Merging data frames and taking transpose of t1 and t2

```
> t1<-merge(dfr1,dfr2)
> t2<-t(t1)
> t2
       [,1]         [,2]
type   "fake"       "real"
pos_sd "0.07686965" "0.05950819"
neg_sd "0.06038243" "0.04470498"
> #merging dataframes and taking transpose of t4 we get t3
> t3<-merge(dfr4,dfr3)
> t4<-t(t3)
> t4
        [,1]         [,2]
type    "fake"       "real"
pos_med "0.04143646" "0.03591160"
neg_med "0.02570694" "0.02313625"
> df2$negative<-normalize(df2$negative)
> df2$positive<-normalize(df2$positive)
> #Combine this with the news dataset
> test<-cbind(test,df2)
```

#finding standard deviations and median of negative and positive columns of each type of news

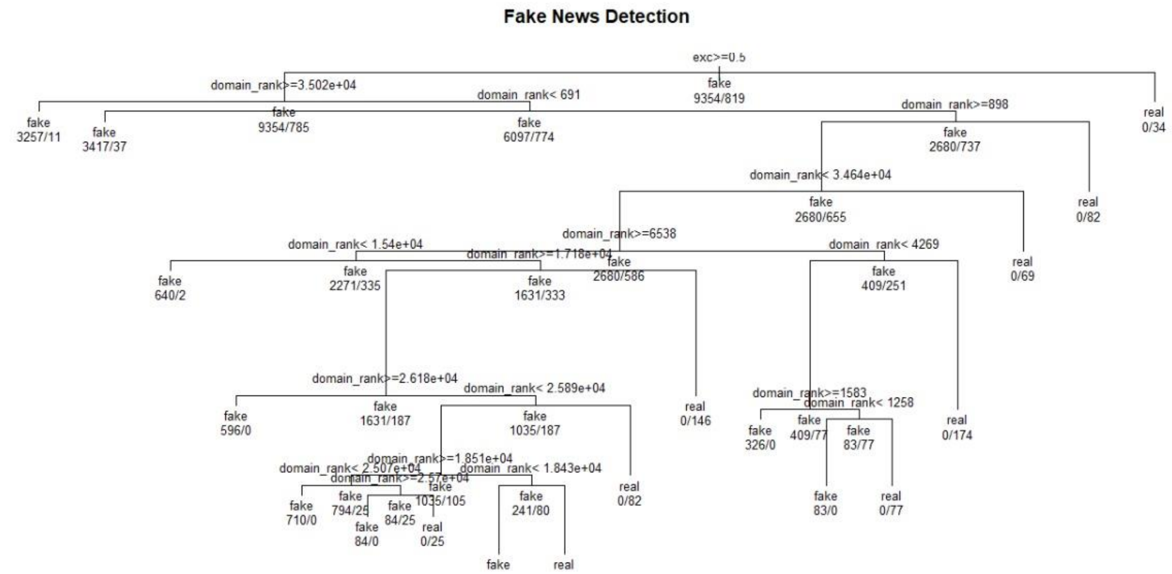
```
> neg_sd1<-test %>% group_by(type) %>% summarise(neg_sd=sd(negative))
> pos_sd1<-test %>% group_by(type) %>% summarise(pos_sd=sd(positive))
> neg_med1<-test %>% group_by(type) %>% summarise(neg_med=median(negative))
> pos_med1<-test %>% group_by(type) %>% summarise(pos_med=median(positive))
> #create dataframes for negative and positive standard deviations and median
> dfr5<-data.frame(neg_sd1)
> dfr6<-data.frame(pos_sd1)
> dfr7<-data.frame(neg_med1)
> dfr8<-data.frame(pos_med1)
```

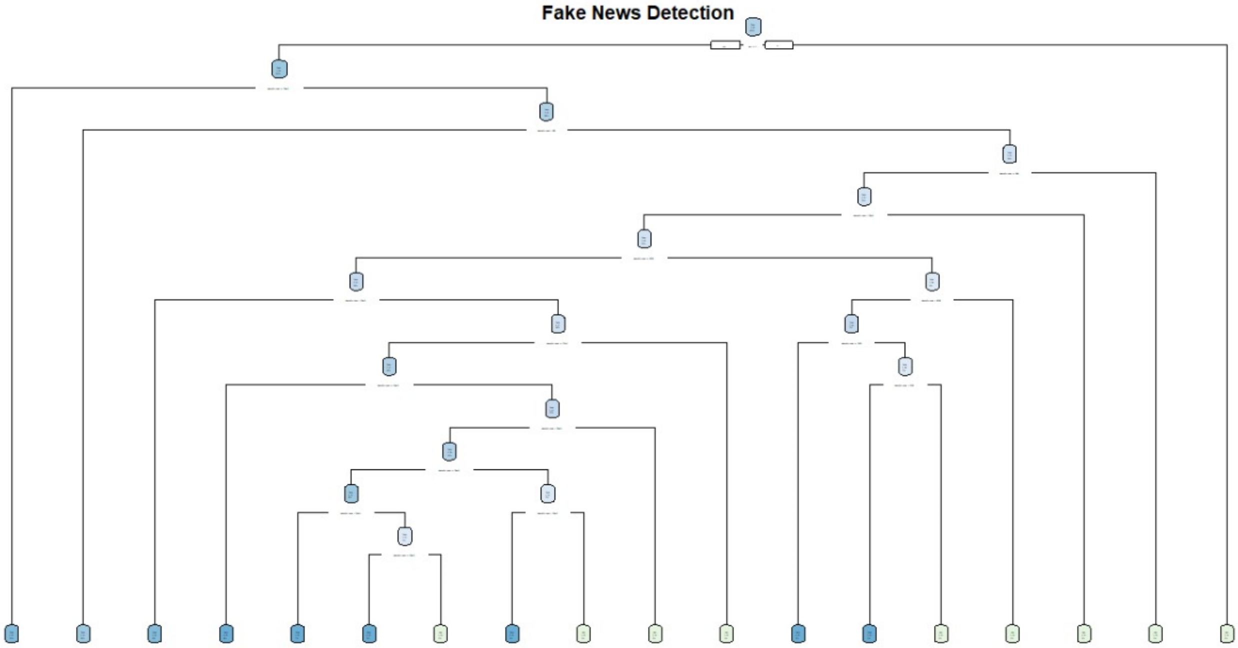
#Merging data frames and taking transpose of t1 to get t2

```
> t5<-merge(dfr5,dfr6)
> t6<-t(t5)
> t6
       [,1]         [,2]
type   "fake"       "real"
neg_sd "0.09089292" "0.06908706"
pos_sd "0.09593252" "0.06942864"
> #merging dataframes and taking transpose of t4 we get t3
> t7<-merge(dfr7,dfr8)
> t8<-t(t7)
> t8
        [,1]         [,2]
type    "fake"       "real"
neg_med "0.03984064" "0.03984064"
pos_med "0.05434783" "0.05072464"
> mod1 = rpart(type ~ domain_rank + spam_score + exc + que + positive + negative
+              , train,method = "class")
> plot(mod1)
> text(mod1, use.n = T, all = T, cex = 0.8)
> treepred = predict(mod1,test,type = "class")
> table(treepred,test$type)

treepred fake real
    fake 2594    7
    real    0  225
> mean(treepred == test$type)
[1] 0.997523
```

DECISION TREE:




Fake News Detection

CONCLUSION:

1. We performed detailed exploratory data analysis on the real and fake news datasets. We generated multiple plots of all variables for both news categories.
2. We analyzed unigrams and bigrams and get some interesting words and phrases which are associated with fake news and included in the title or body of the news.
3. There are some common words and phrases which might be associated with a particular type of news report and might be used to manipulate the language of the title or body of news.
4. And we also used a machine learning model decision tree and we found its accuracy.

REFERENCES:

- Alonso, M.; Vilares, D.; Gómez-Rodríguez, C.; Vilares, J. Sentiment Analysis for Fake News Detection. Electronics 2021, 10, 1348. [CrossRef]
- Rehma, A.A.; Awan, M.J.; Butt, I. Comparison and Evaluation of Information Retrieval Models. VFAST Trans. Softw. Eng. 2018, 13, 7–14. [CrossRef]
- Alam, T.M.; Awan, M.J. Domain analysis of information extraction techniques. Int. J. Multidiscip. Sci. Eng. 2018, 9, 1–9.
- Kim, H.; Park, J.; Cha, M.; Jeong, J. The Effect of Bad News and CEO Apology of Corporate on User Responses in Social Media. PLoS ONE 2015, 10, e0126358. [