ILLINOIS INSTITUTE OF TECHNOLOGY

FAKE NEWS DETECTION

Niveditha Mangala Venkatesha

A20466182

(Email id: nmangalavenkatesha@hawk.iit.edu)

May 1st, 2022

Prof. Lulu Kang

Statistical Learning MATH 569

ABSTRACT

We live in a modern world in which data is essential, and 1.7 megabytes of data are generated every second. Due to this large amount of data, we have many technologies that impact our lives. The internet is one of the most essential innovations, and it is used by a vast number of people. These individuals have access to a variety of social media channels. These online platforms allow any user to create a post or share information. Users or their posts are not verified on these networks. As a result, some people attempt to distribute fake news using these channels. Fake news can be used to spread propaganda against a person, a society, an organization, or a political party. The media is seen as a powerful force in a variety of disciplines. Though the media alone has a significant impact on society, additional influence from well-known parties can lead to indirect mass manipulation. All of this bogus news is impossible to detect by a human. As a result, regression models that can detect bogus news automatically are used in this project.

INTRODUCTION

Digital technology has several advantages as well as disadvantages. One problem is fake news. It is easy for someone to spread false information. The most basic definition of fake news is information that leads people astray. Fake news is spreading like wildfire nowadays, and people are sharing it without confirming it. This is frequently done to promote or impose specific views, and it is frequently accomplished through political agendas. To produce online advertising revenue, media outlets must be able to draw viewers to their websites. As mobile devices and internet access become more widespread, fake news hurts society.

As a result of the pandemic, state officials have turned to social media platforms, which could have a severe impact on local state integrity due to the proliferation of bogus news. Users of social media, ranging from minority groups or people to well-known personalities and influencers, are all vulnerable to the risk of information fabrication or fake news perpetrated by uninformed users. When the masses are bombarded with information, they prefer influenced insights over profound sources, even famous people can spread disinformation by being more appealing. Social media is one of the most common platforms for disseminating false news and badly written news pieces that may contain some truth but aren't entirely accurate or entirely made up.

Typically, this type of news is intended to promote or defend a particular agenda or point of view. One cause for the rise in misleading information is the simplicity with which anyone can publish false reviews or articles on the web, with no pre-approval process in place, and then propagate erroneous information or beliefs throughout all social media platforms. Many readers will find it difficult to determine if the content is false or not. As a result, it is critical to detect fake news [1].

The main challenge in this line of research is collecting quality data, i.e., instances of fake and real news articles on a balanced distribution of topics. In this assignment, I will use a dataset from Kaggle that has already separated a collection of articles into Fake or True. I will be training an algorithm to detect if a new article is false or true by simply using multiple regressions gathered from the text of the articles. Regression is a model that assesses the relationship between a dependent variable and an independent variable by fitting a line to the observed data. Using different methodologies like logistic regression, support vector machine

(SVM), decision tree, and random forest models using training and testing data sets to get an accuracy of the true news.

DATA SOURCES

The dataset used for the project is from Kaggle's dataset list. Source-based Fake News classification fake and real news dataset. The given dataset-1 is:

Fake News Data:

```
## # A tibble: 6 x 4
##   title                          text                           subject date
##   <chr>                          <chr>                          <chr>   <chr>
## 1 " Donald Trump Sends Out Emba~ Donald Trump just couldn t w~ News    December~
## 2 " Drunk Bragging Trump Staffe~ House Intelligence Committee~ News    December~
## 3 " Sheriff David Clarke Become~ On Friday, it was revealed t~ News    December~
## 4 " Trump Is So Obsessed He Eve~ On Christmas day, Donald Tru~ News    December~
## 5 " Pope Francis Just Called Ou~ Pope Francis used his annual~ News    December~
## 6 " Racist Alabama Cops Brutali~ The number of cases of cops ~ News    December~
```

True News Data:

```
## # A tibble: 6 x 4
##   title                      text                           subject  date
##   <chr>                      <chr>                          <chr>    <chr>
## 1 As U.S. budget fight looms~ "WASHINGTON (Reuters) - The h~ politic~ "December~
## 2 U.S. military to accept tr~ "WASHINGTON (Reuters) - Trans~ politic~ "December~
## 3 Senior U.S. Republican sen~ "WASHINGTON (Reuters) - The s~ politic~ "December~
## 4 FBI Russia probe helped by~ "WASHINGTON (Reuters) - Trump~ politic~ "December~
## 5 Trump wants Postal Service~ "SEATTLE/WASHINGTON (Reuters)~ politic~ "December~
## 6 White House, Congress prep~ "WEST PALM BEACH, Fla./WASHIN~ politic~ "December~
```

There are a total of 23481 rows of fake news data and 21417 rows of true news data. To have a better outcome, I will be using the title and article text to identify whether an article is fake or true news will be determined.

From the above details I have observed so far:

- There are more fake news articles than true news articles
- Title and text fields are joined into one text field
- The sources of articles and authors are unknown
- Date and subject fields are not required for the above analysis so these 2 fields are being removed
- I will be adding a Fake or True flag when combining data

The next dataset-2 is the new article's data, where the table is shown below,

```
## # A tibble: 6 x 12
##   author  published  title   text    language site_url main_img_url  type  label
##   <chr>   <chr>      <chr>   <chr>   <chr>    <chr>    <chr>         <chr> <chr>
## 1 Barrac~ 2016-10-2~ muslim~ print ~ english  100perc~ http://bb4sp~ bias  Real
## 2 reason~ 2016-10-2~ attorn~ attorn~ english  100perc~ http://bb4sp~ bias  Real
## 3 Barrac~ 2016-10-3~ breaki~ red st~ english  100perc~ http://bb4sp~ bias  Real
## 4 Fed Up  2016-11-0~ pin dr~ email ~ english  100perc~ http://100pe~ bias  Real
## 5 Fed Up  2016-11-0~ fantas~ email ~ english  100perc~ http://100pe~ bias  Real
## 6 Barrac~ 2016-11-0~ hillar~ print ~ english  100perc~ http://bb4sp~ bias  Real
## # ... with 3 more variables: title_without_stopwords <chr>,
## #   text_without_stopwords <chr>, hasImage <int>
```

In the new dataset, I have observed that there is a new column type called, “type” which would be a great predicting factor as this column does not contain this information in dataset-1. To remove bias, I have removed all other columns except the title and text. I have concatenated true news data from both datasets 1 and 2. I have added only the true articles from Data set 2 and ensured that there is an equal amount of Fake and True articles. So that there is no bias when predicting. Changing the label of Real to True articles and removing all other columns except title & text. And there are a total of 22218 data rows for both fake news and true news datasets.

PROPOSED METHODOLOGY

DATA PREPROCESSING

1. ANALYSIS OF THE WORDS

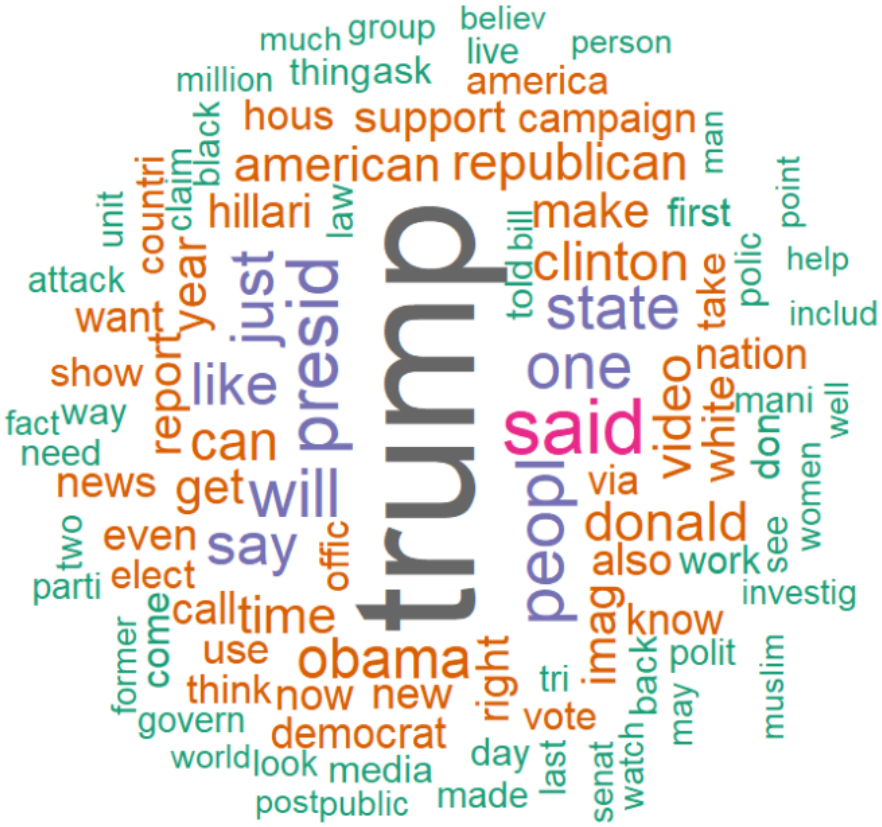
For the analysis of the words, I have used text preprocessing logic which is explained below:

- Finding the most common words in fake & true.
- Add title & text into the same field.
- Loading the text as a corpus text means it is a language resource consisting of a large and structured set of texts.
- Have converted the text to lower case
- Have removed numbers and punctuations
- Removed English common ‘stopwords’ which means they are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence.
- Eliminating extra white spaces

- Text stemming has also been applied, that is it reduces words to their root form. Word types or classes are lessened in the data as a result of stemming by returning the words to their original form. In the stemming process, the words "Running," "Ran," and "Runner" will all be reduced to the word "run." Observation: From my research, this is not always the most accurate process since the stemming process sometimes cut off a full word.

After we have processed the text, we will have a word cloud and a table presenting results for both fake news and true news word cloud. Word Clouds are visual representations of words that give greater prominence to words that appear more frequently. For Word Clouds, the words that are added most frequently by audience members using their smartphones.

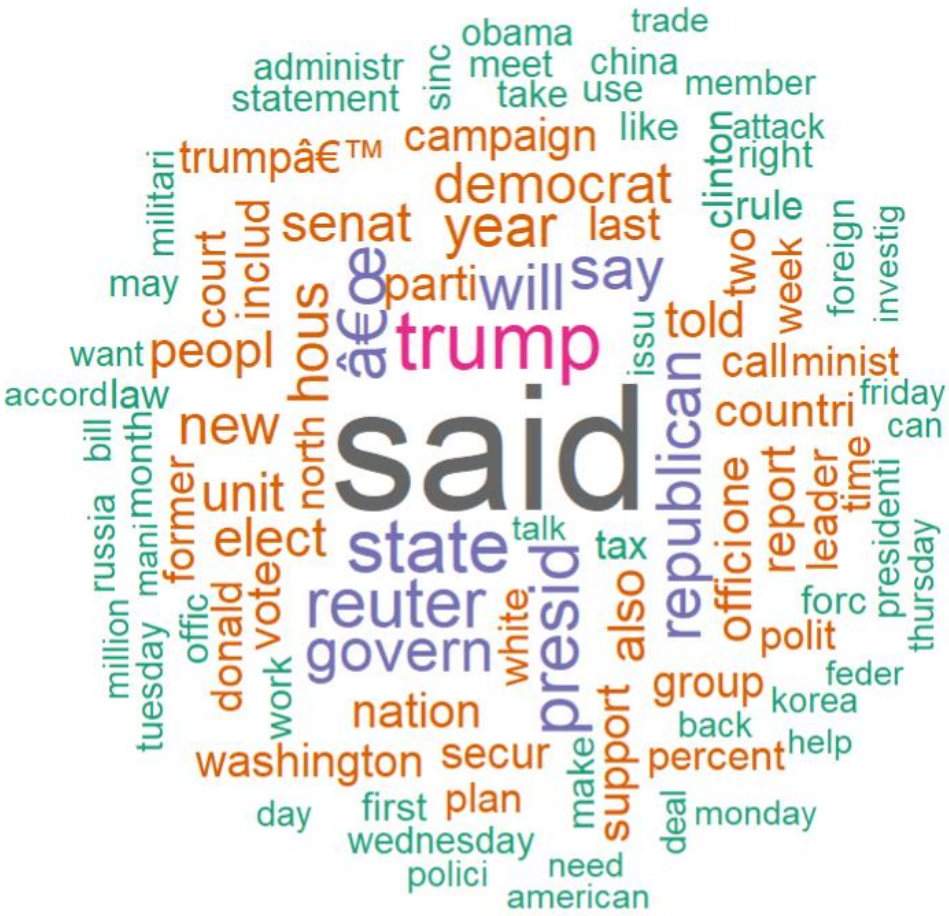
Fake news word cloud:



Fake News: Top5 most Frequent words

	word	freq	type
trump	trump	77693	fake
said	said	29566	fake
presid	presid	27345	fake
peopl	peopl	25112	fake
will	will	24148	fake

I have applied the same text preprocessing logic for true news word data and obtained the True news word cloud is as shown below:



True News: Top5 most Frequent words

	word	freq	type
said	said	99910	true
trump	trump	49413	true
state	state	36651	true
presid	presid	28406	true
reuter	reuter	28306	true

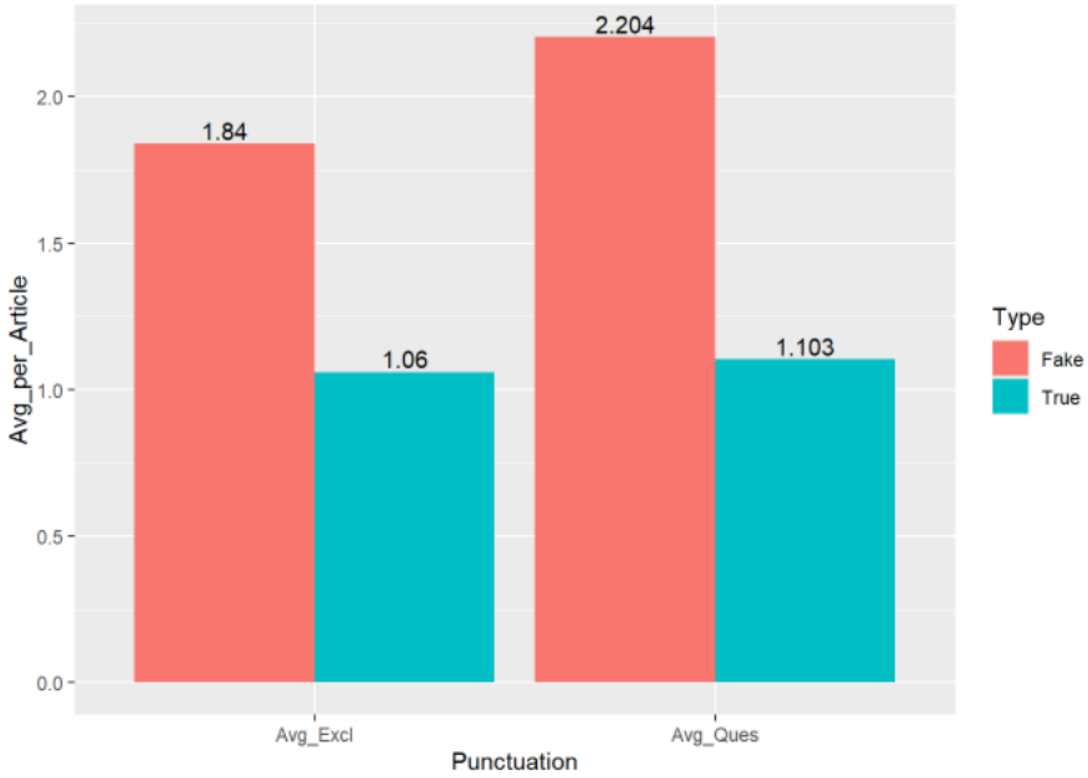
2. TRANSFORMATION OF THE DATA

After creating the word cloud for fake news and true news, I am adding a fake and truth indicator for the prediction process and combining them into a new data variable 'NewsData'. Then the following transformation is applied to the 'NewsData':

- Checking for any "null values" even in the columns section and removing the NA values
- Adding an ID field to the data and removing the 'Title' column as it was already in the data
- Adding the number of sentences per article
- Added number of characters per article

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      15    1304    2265    2487    3173   49766
```

- Detection of the difference between fake news and true news through the punctuations used in the news data. We calculate the number of exclamation marks and question marks using the Sapply function.



The above graph shows the number of exclamatory and question mark punctuations in the fake news and true news. There is more punctuation in false news than in true news. This may be directly related to the number of words, sentences, or the length of the article.

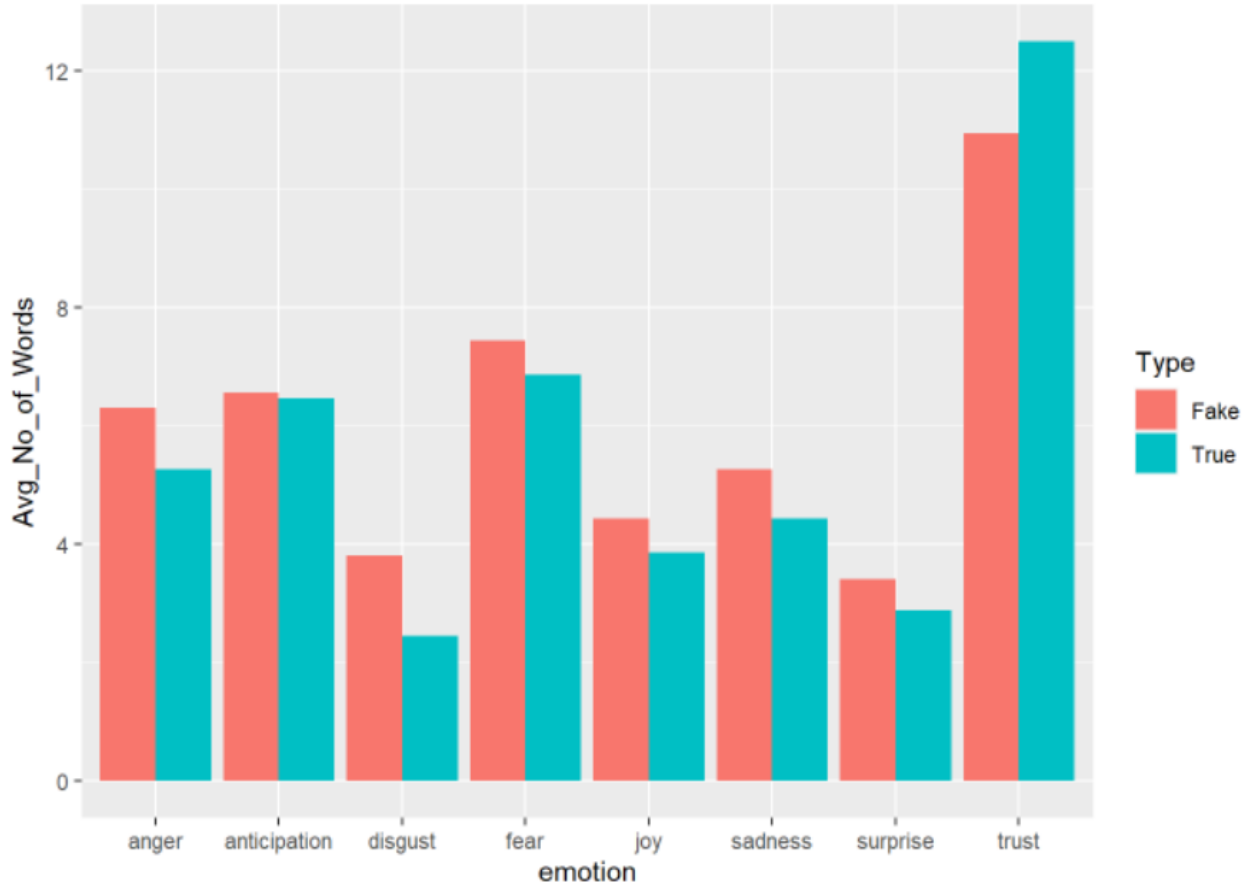
- Making all texts in lower case
- Adding the number of times, the word 'trump' and 'said' have been found in the news data set

Data Measures by article type

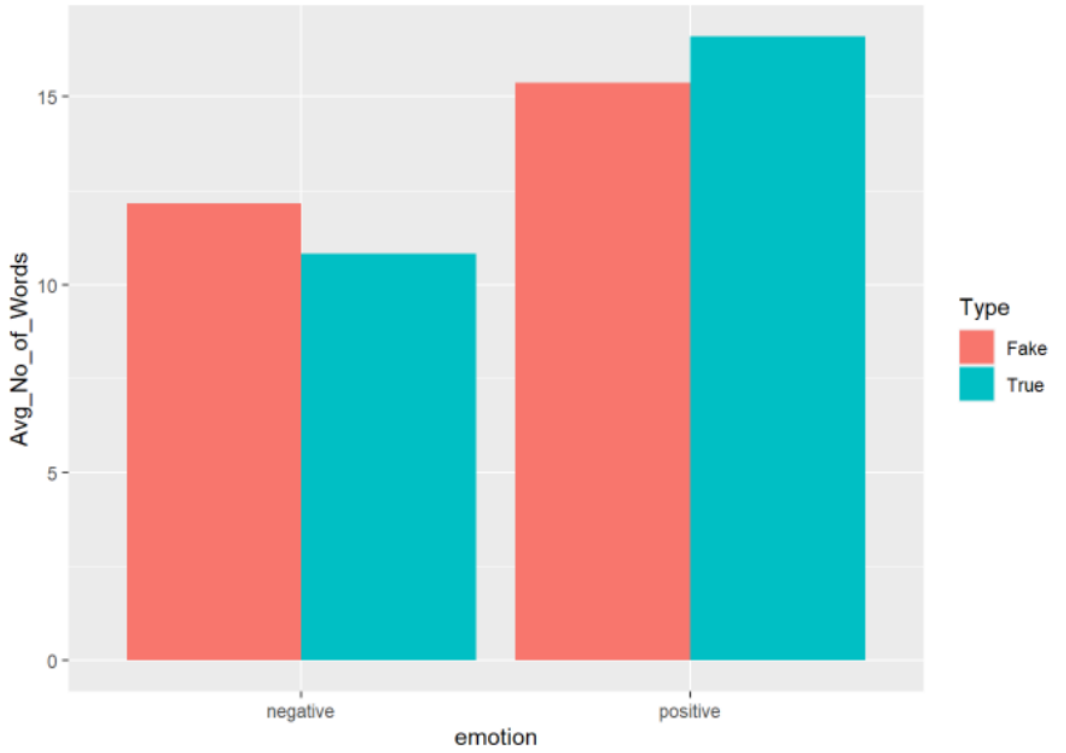
Type	No_articles	Avg_no_Sentences	Avg_TextLength	Avg_no_excl	Avg_no_question	Avg_no_trump	Avg_no_said
Fake	22218	15.57296	2519.192	1.839544	2.203844	4.120623	1.451346
True	22218	13.90904	2454.817	1.059636	1.103205	2.819651	4.505671

- Removing stop words: When analyzing the text, stop words (most common words in a language that do not provide much context) should be removed because they are more common and offer less useful information. Before performing further analysis, they should be removed. For example, conjunctions "and", "or" and "but", prepositions like "of", "in", "from", "to", and the articles "a", "an", and "the"

- As a result of removing the stop words, there are many white spaces between words, so will be removing the duplicate white spaces and adding several words per article after removing stop words
- I am doing sentiment analysis on the news data set which extracts information to identify reactions, attitudes, context, and emotions. Expressions like anger, fear, anticipation, trust, surprise, sadness, joy, and disgust are obtained from the 'newsdata' set of this project [2].



Among the sentiments measured, trust is the only one that is higher for true news than for fake news. I will include this observation in my working dataset [3].



For this analysis, the negative, positive, and trust will be used as predictors.

DATA MODELS

4 different types of data models have been implemented to predict if an article is true or false using the following models:

1. Logistic regression:

   Logistic regression is used in statistical software to estimate probabilities and understand the relationship between the dependent variable and one or more independent variables using a logistic regression equation. This type of analysis can assist you in predicting the likelihood of an event occurring or a decision being made. I have created 2 different models using logistic regression, one with just only the number of sentences and text length for the prediction. And the other one uses the number of words and sentiment of the news.

2. Support vector machine:

   Support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. The support vector machine algorithm's goal is to find a hyperplane in N-dimensional space (N - the number of features) that classifies the data points. Even SVM has used different

inputs in 2 different models. In the first model, I am using just several sentences and text length columns for the prediction and in the other one, I am using all the fields for the prediction.

3. Decision tree:

   A decision tree is a tree-like model that serves as a decision-making aid by visually displaying decisions and their potential outcomes, consequences, and costs. The "branches" can then be easily evaluated and compared to select the best courses of action. We are using all fields to find the accuracy of the news.

4. Random forest:

   Random forests or random decision forests is an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

ANALYSIS AND RESULTS

After we have prepared the data, we can begin building and training the data. Because the majority of data models only accept numerical variables, categorical variables must be preprocessed. We must convert these categorical variables to numbers so that the model can understand and extract useful data. Fake and True will be set to 1 and 0 in this case. I have split the dataset into the training set and test set.

Before creating the models, feature scaling is required. Feature scaling is a technique for normalizing a set of independent variables or data components. It's also known as data normalization in the data processing.

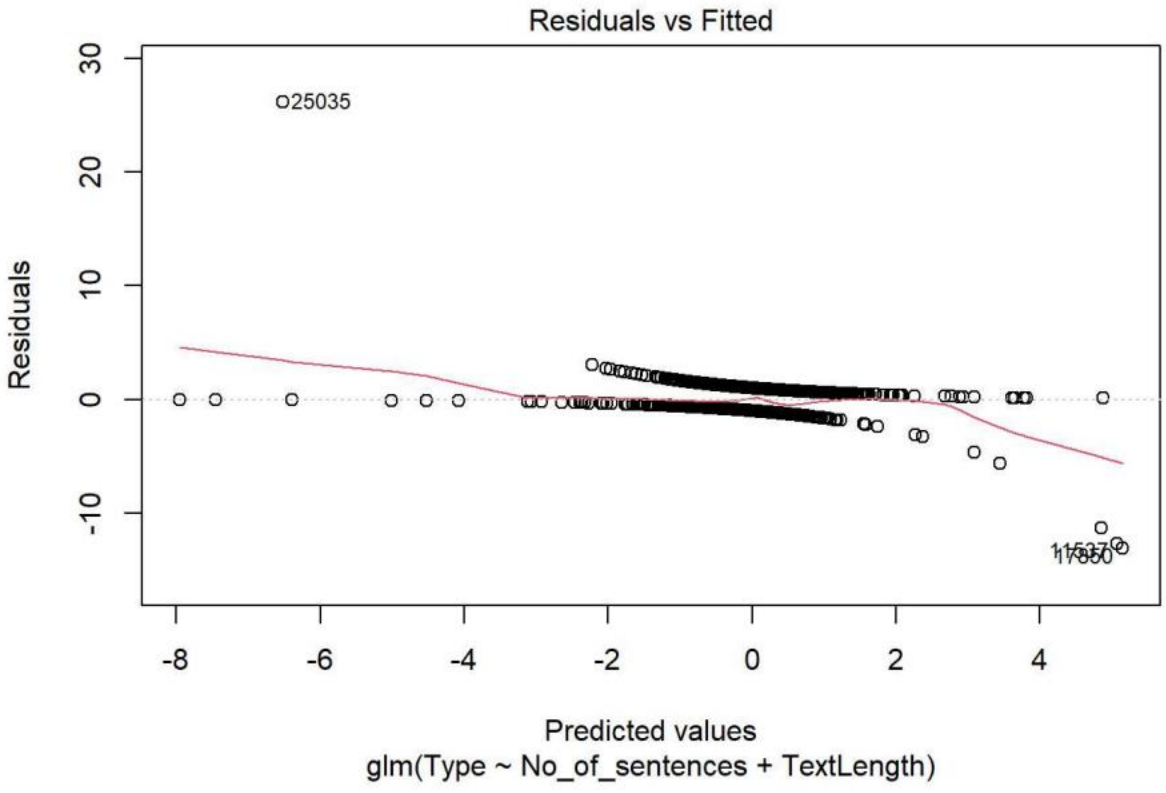
Model 1: Logistic regression

The number of Sentences and Text Length will be used for prediction in the first model.

- Fitting logistic regression to the training set

```
##
## Call:  glm(formula = Type ~ No_of_sentences + TextLength, family = binomial,
##     data = train_set)
##
## Coefficients:
##     (Intercept)  No_of_sentences       TextLength
##      -0.0008657       -0.5062275        0.4085171
##
## Degrees of Freedom: 35547 Total (i.e. Null);  35545 Residual
## Null Deviance:       49280
## Residual Deviance: 48810     AIC: 48820
```

- The plot between residuals vs fitted for the model-1 is shown below.



- A confusion matrix is an excellent tool for calibrating a model's output and assessing all of the probable outcomes of your predictions (true positive, true negative, false positive, false negative). And the confusion matrix and statistics of model-1 are given below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1583 2861
##          1 2066 2378
##
##                Accuracy : 0.4457
##                  95% CI : (0.4353, 0.4561)
##     No Information Rate : 0.5894
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.1087
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4338
##             Specificity : 0.4539
##          Pos Pred Value : 0.3562
##          Neg Pred Value : 0.5351
##              Prevalence : 0.4106
##          Detection Rate : 0.1781
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.4439
##
##        'Positive' Class : 0
##
```

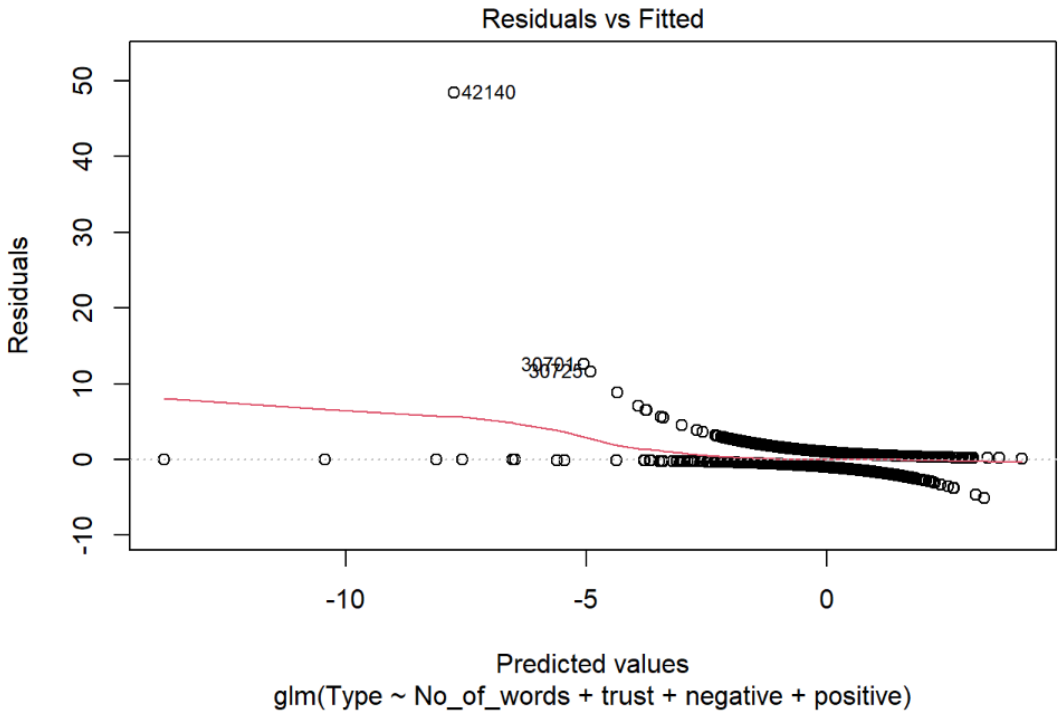
- We can see from the graph that using simply the number of sentences and the text length as predictors, with an accuracy of 45%, is not that suitable for prediction. So, let's examine how these changes when it is checked with different predictors.

Model 2: Logistic regression 2

With the sentiment of the phrases and number of words, I'm using Logistic Regression once more. And fitting logistic regression to the training set gives,

```
##
## Call:  glm(formula = Type ~ No_of_words + trust + negative + positive,
##     family = binomial, data = train_set)
##
## Coefficients:
## (Intercept)  No_of_words        trust     negative     positive
##  -9.374e-05   -8.805e-01    9.996e-01   -3.854e-01    2.648e-01
##
## Degrees of Freedom: 35547 Total (i.e. Null);  35543 Residual
## Null Deviance:       49280
## Residual Deviance: 46590     AIC: 46600
```

The plot between residuals vs fitted for model-2 is shown below.



After predicting training set data with the test set results, I see that the accuracy for model-2 has been reduced to 37.56%. And I am using a different method of prediction.

Model 3: Support Vector Machine

In this model, I am using only the number of sentences and text length, for fitting the training set to the support-vector machine model and predicting the test set results as:

```
##
## Call:
## svm(formula = Type ~ No_of_sentences + TextLength, data = train_set,
##     type = "C-classification", kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  34163
```

While predicting the test results, I have obtained the confusion matrix and statistics of the model-3 which are given below,

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1 1234  759
##          0 3210 3685
##
##                Accuracy : 0.5534
##                  95% CI : (0.543, 0.5638)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.1069
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.2777
##             Specificity : 0.8292
##          Pos Pred Value : 0.6192
##          Neg Pred Value : 0.5344
##              Prevalence : 0.5000
##          Detection Rate : 0.1388
##    Detection Prevalence : 0.2242
##       Balanced Accuracy : 0.5534
##
##        'Positive' Class : 1
##
```

Accuracy of the article if it is true or fake is 55% which is higher than all models of logistic regression.

Model 4: Support Vector Machine 2

In this Algorithm, I use all of the fields to predict whether an article is True or False, and I fit SVM to the Training set.

```
##
## Call:
## svm(formula = Type ~ ., data = train_set, type = "C-classification",
##     kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  13763
```

While forecasting the test results, I received the confusion matrix and statistics of model-3, which are shown below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1 3549  414
##          0  895 4030
##
##                Accuracy : 0.8527
##                  95% CI : (0.8452, 0.86)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7054
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7986
##             Specificity : 0.9068
##          Pos Pred Value : 0.8955
##          Neg Pred Value : 0.8183
##              Prevalence : 0.5000
##          Detection Rate : 0.3993
##    Detection Prevalence : 0.4459
##       Balanced Accuracy : 0.8527
##
##        'Positive' Class : 1
##
```

```
## Accuracy
##     0.85
```

The accuracy of model-4 for the support-vector machine is 85% which is higher than all models we have seen so far.

Model 5: Decision Tree

All the fields of the 'newsData' set have been used in this model, and fitting the decision tree [5] to the training set is,

```
## n= 35548
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 35548 17774 1 (0.50000000 0.50000000)
##    2) No_of_question>=1.5 10106  1168 1 (0.88442509 0.11557491) *
##    3) No_of_question< 1.5 25442  8836 0 (0.34729974 0.65270026)
##      6) No_of_Wordsaid< 0.5 5782  1424 1 (0.75371844 0.24628156) *
##      7) No_of_Wordsaid>=0.5 19660  4478 0 (0.22777213 0.77222787)
##       14) No_of_excl>=1.5 1762   444 1 (0.74801362 0.25198638) *
##       15) No_of_excl< 1.5 17898  3160 0 (0.17655604 0.82344396)
##         30) No_of_Wordsaid< 1.5 3537  1288 0 (0.36415041 0.63584959)
##           60) No_of_words>=117.5 2048   817 1 (0.60107422 0.39892578) *
##           61) No_of_words< 117.5 1489    57 0 (0.03828073 0.96171927) *
##         31) No_of_Wordsaid>=1.5 14361  1872 0 (0.13035304 0.86964696) *
```

In the decision tree model, after predicting the test set results, I am getting the accuracy of whether the article is fake or not is 84%.

Model 6: Random Forest

'NewsData' sets' all fields have been used for splitting into a training set and test set and fitting the training set for the random forest model [4].

```
##
## Call:
##  randomForest(x = train_set[-1], y = train_set$Type, ntree = 500,      mtry =
6, localImp = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 7.86%
## Confusion matrix:
##       1     0 class.error
## 1 16510  1264  0.07111511
## 0  1529 16245  0.08602453
```

For the random forest model, I have found an accuracy of 90.991%, which is higher than all other models used in the project.

The above comparison demonstrates the true power of ensembling as well as the significance of using Random Forest over Decision Trees. Though Random Forest has its restrictions, such as the number of factor levels a categorical variable can have, it is still one of the strongest models for categorization. Random Forest maintains a high level of accuracy.

CONCLUSION

Despite the model's high accuracy of 90.9 percent, this is a small sample dataset. For the text, I simply used a limited number of measures. To expand on this study and analysis, it would be useful to collect data from numerous sources and on various topics, as well as to include alternative dimensions such as Author, News Channel, Website, Topic, Country or Region, and so on. This can significantly increase the usefulness of fake checking detection. Word associations, term frequencies, and phases, as well as stemming, can all be used to help with fake news analysis.

REFERENCES

[1] https://medium.com/swlh/fake-news-detection-using-machine-learning-69ff9050351f

[2] https://datascienceplus.com/parsing-text-for-emotion-terms-analysis-visualization-using-r/

[3] https://medium.com/swlh/exploring-sentiment-analysis-a6b53b026131

[4] https://cran.r-project.org/web/packages/randomForestExplainer/vignettes/randomForestExplainer.html

[5] https://www.tutorialspoint.com/r/r_decision_tree.htm#