QTM2000 Name: Angelina Cho

Case Studies in Business Analytics

Professor Mathaisel

Fall 2021

Exam 2

Individual Assignment

The following pledge must be on your exam cover sheet and signed.

I pledge my honor that I have neither received nor provided unauthorized assistance during the completion of this work. Please Initial: AC

The objective of this project is to predict whether or not news is fake or real based on the text characteristics/attributes in the text. This type of prediction is known in analytics as classification. There are numerous techniques and models for classification, and the project will explore some of these techniques. For this project, you are being asked to write an "Executive Summary" to accomplish that goal using the dataset provided and its accompanying R-script. Your summary should be brief, but it must include: an Introduction, Analysis, and Conclusion. The analysis must address three parts: 1. Data "Wrangling" and Visualizations; 2. Naive Bayes; and 3. Logistic Regression.

Rubric:

Part 1.	30 points
Part 2.	35 points
Part 3.	35 points
	100 points

Introduction

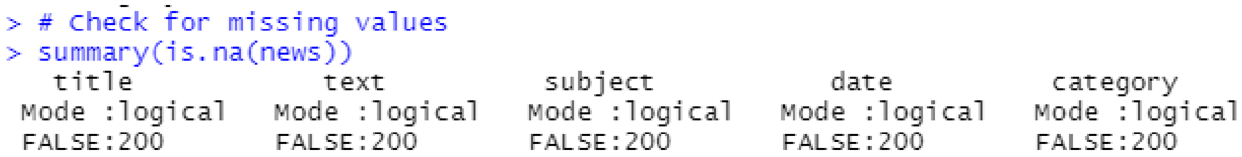
News can impact the political and economic environment by influencing people's emotions. Therefore, it is important to distinguish between fake news and true news. The purpose of this executive summary is to build a logistic regression model and a Naive Bayes Model to predict whether the news is fake or true based on the characteristics of the newspaper. The evaluation of the data will be a random sample of 200 observations in total from each of the large datasets of Fake dataset (3209 observations, 4 variables) and True dataset (6556 observations, 4 variables). The two datasets are then merged into one for the purpose of classification modelling. The specific attributes of the Real or Fake News include the title, text, subject, date, and category. The attribute, "title", is the headline that catches the attention of readers and relates well to the news topic. The attribute, "text", is the body of the article. The attribute, "subject", indicates the subject of the news article. The attribute, "date", refers to the date of the news. The 5th variable is Category, which is manually assigned to the observations. "0" is assigned to fake news observations as a category variable, and "1" is assigned to true news observations as a category variable. It will then be split into training and testing sets, using the 75/25 rule. The datasets are imported from Buzzfeed news a week close to the 2016 U.S. Presidential Election from September 19 to 23, 26, and 27. All the articles were fact-checked by five BuzzFeed journalists and sorted into one dataset of fake news and another dataset of real news in the form of csv files.

The datasets themselves are text data related to the 2016 U.S. Presidential Election. Text is similar to categorical data, but more related to the human language. Therefore, we will be using Natural Language Processing (NLP) to do some data cleaning/wrangling to our data before we start the classification processes. We will also be transforming the data into factor data types to generate the visualizations, for example, Figure 2 (Category variable changed to factor data type) and Figure 3 (Subject variable changed to factor data type).

Naive Bayes Model is a classification algorithm based on Bayes' Theorem with an assumption that the presence of a particular feature of a class is unrelated to the presence of any other feature. It only works when all of the predictor variables are categorical (factor). Therefore, we will, like we've talked about previously, transform the data into factor data types in order to classify them. The objective is to predict an unknown binary response (fake news or real news) from the predictor variables, which are assumed categorical. Based on the conditional probabilities, we assume that the value of a particular predictor (Category variable) is independent of the value of any other predictor, given the class. It assigns the new observation to the class which had the highest probability.

Logistic regression is a predictive modeling algorithm that is used when the predictive (Y) variable is binary categorical. In this case, 0 is assigned, and represents, fake news in the category variable, and 1 is assigned, and represents, true news in the category variable.

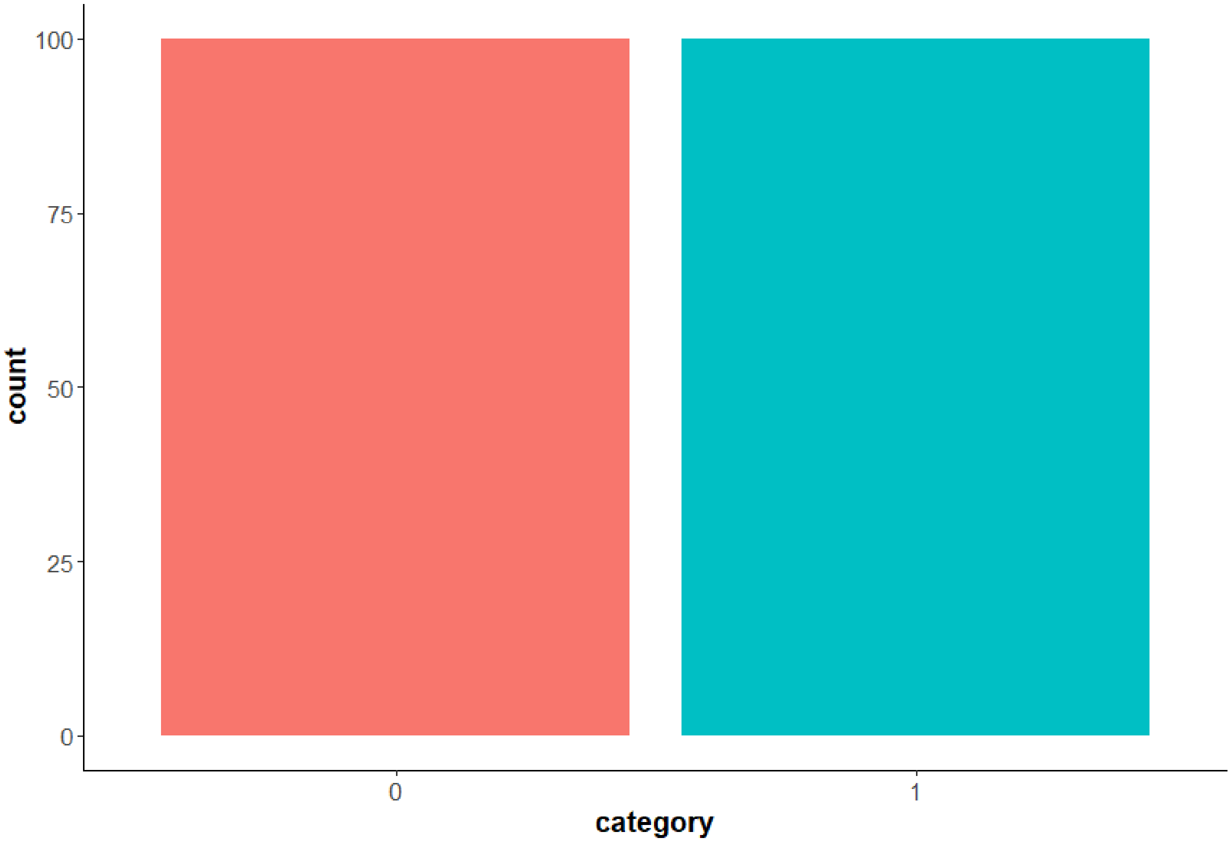
Figure 1 Check for Missing Values



```
> # check for missing values
> summary(is.na(news))
   title             text            subject             date            category      
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical  
 FALSE:200       FALSE:200       FALSE:200       FALSE:200       FALSE:200      
```

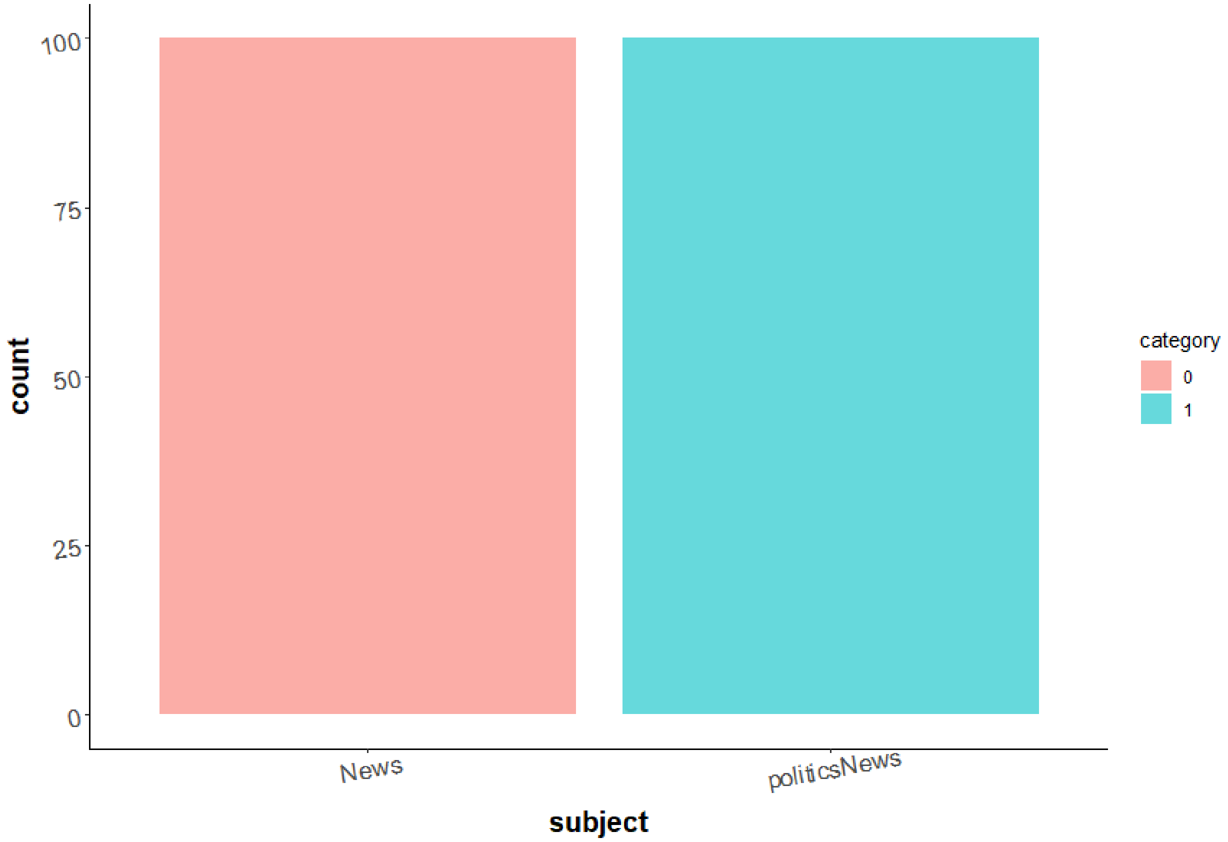
Figure 1 checks for any missing values in the 5 variables. There is, clearly, no missing values in the datasets because all five variables show “FALSE: 200”, meaning that there are no missing values out of all 200 observations.

Figure 2 Bar Graph showing the number of observations by Category



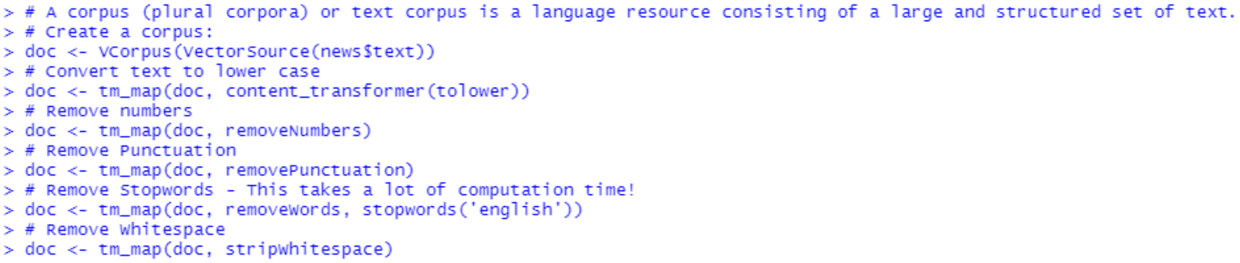
This figure shows the number of observations we have for each Category (0: fake news, 1: true/real news). Since we only took a random sample of 100 observations from the Fake dataset and 100 observations from the True dataset, we ended up having 200 observations in total, as shown in Figure 2.

Figure 3 Bar Graph showing the number of observations by Subject and Category

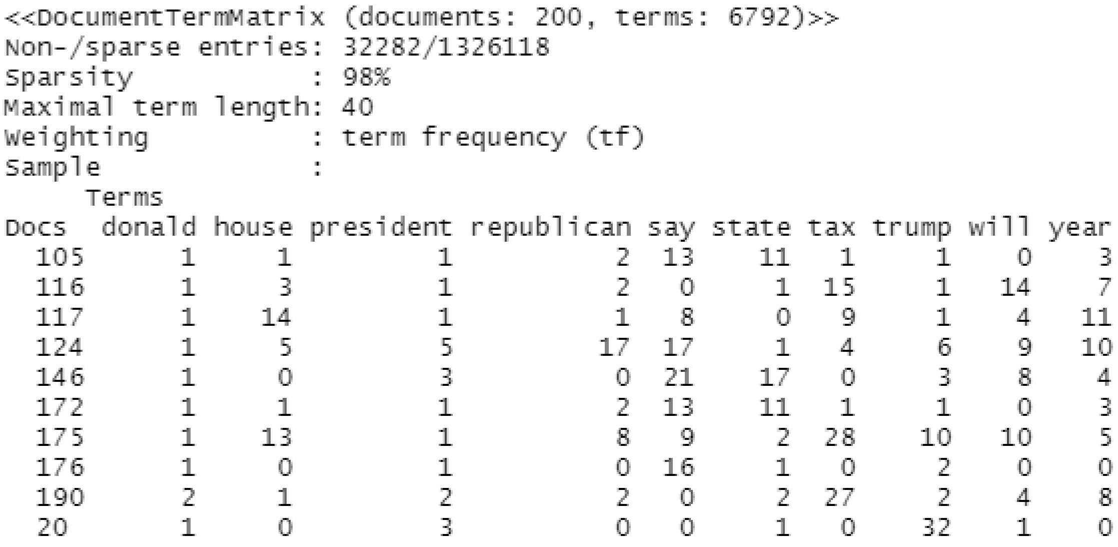


This figure shows the number of observations we have for each subject, News and Politics News. The number of News observations is 100 and the number of Politics News is also 100. In addition, the number of news that are of Subject News and are fake news is 100, and the number of news that are of Subject Politics News and are real news is 100.

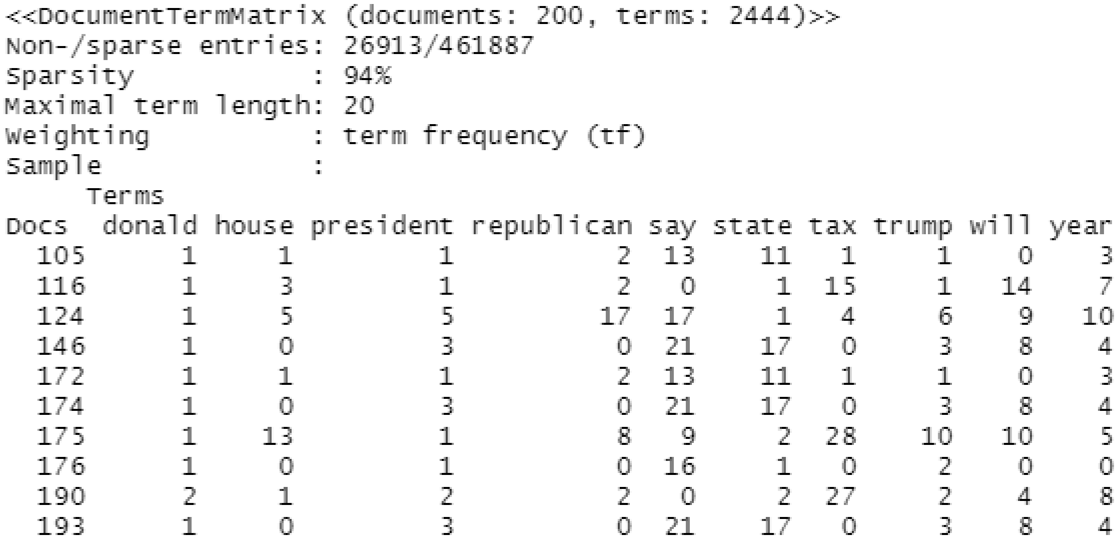
Figure 4.1, 4.2 & 4.3 Data "Wrangling" & Document Term Matrix



```
> # A corpus (plural corpora) or text corpus is a language resource consisting of a large and structured set of text.
> # Create a corpus:
> doc <- VCorpus(VectorSource(news$text))
> # Convert text to lower case
> doc <- tm_map(doc, content_transformer(tolower))
> # Remove numbers
> doc <- tm_map(doc, removeNumbers)
> # Remove Punctuation
> doc <- tm_map(doc, removePunctuation)
> # Remove Stopwords - This takes a lot of computation time!
> doc <- tm_map(doc, removeWords, stopwords('english'))
> # Remove Whitespace
> doc <- tm_map(doc, stripWhitespace)
```



```
<<DocumentTermMatrix (documents: 200, terms: 6792)>>
Non-/sparse entries: 32282/1326118
Sparsity           : 98%
Maximal term length: 40
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs  donald house president republican say state tax trump will year
  105      1     1         1          2  13    11   1     1    0    3
  116      1     3         1          2   0     1  15     1   14    7
  117      1    14         1          1   8     0   9     1    4   11
  124      1     5         5         17  17     1   4     6    9   10
  146      1     0         3          0  21    17   0     3    8    4
  172      1     1         1          2  13    11   1     1    0    3
  175      1    13         1          8   9     2  28    10   10    5
  176      1     0         1          0  16     1   0     2    0    0
  190      2     1         2          2   0     2  27     2    4    8
  20       1     0         3          0   0     1   0    32    1    0
```



```
<<DocumentTermMatrix (documents: 200, terms: 2444)>>
Non-/sparse entries: 26913/461887
Sparsity           : 94%
Maximal term length: 20
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs  donald house president republican say state tax trump will year
  105      1     1         1          2  13    11   1     1    0    3
  116      1     3         1          2   0     1  15     1   14    7
  124      1     5         5         17  17     1   4     6    9   10
  146      1     0         3          0  21    17   0     3    8    4
  172      1     1         1          2  13    11   1     1    0    3
  174      1     0         3          0  21    17   0     3    8    4
  175      1    13         1          8   9     2  28    10   10    5
  176      1     0         1          0  16     1   0     2    0    0
  190      2     1         2          2   0     2  27     2    4    8
  193      1     0         3          0  21    17   0     3    8    4
```

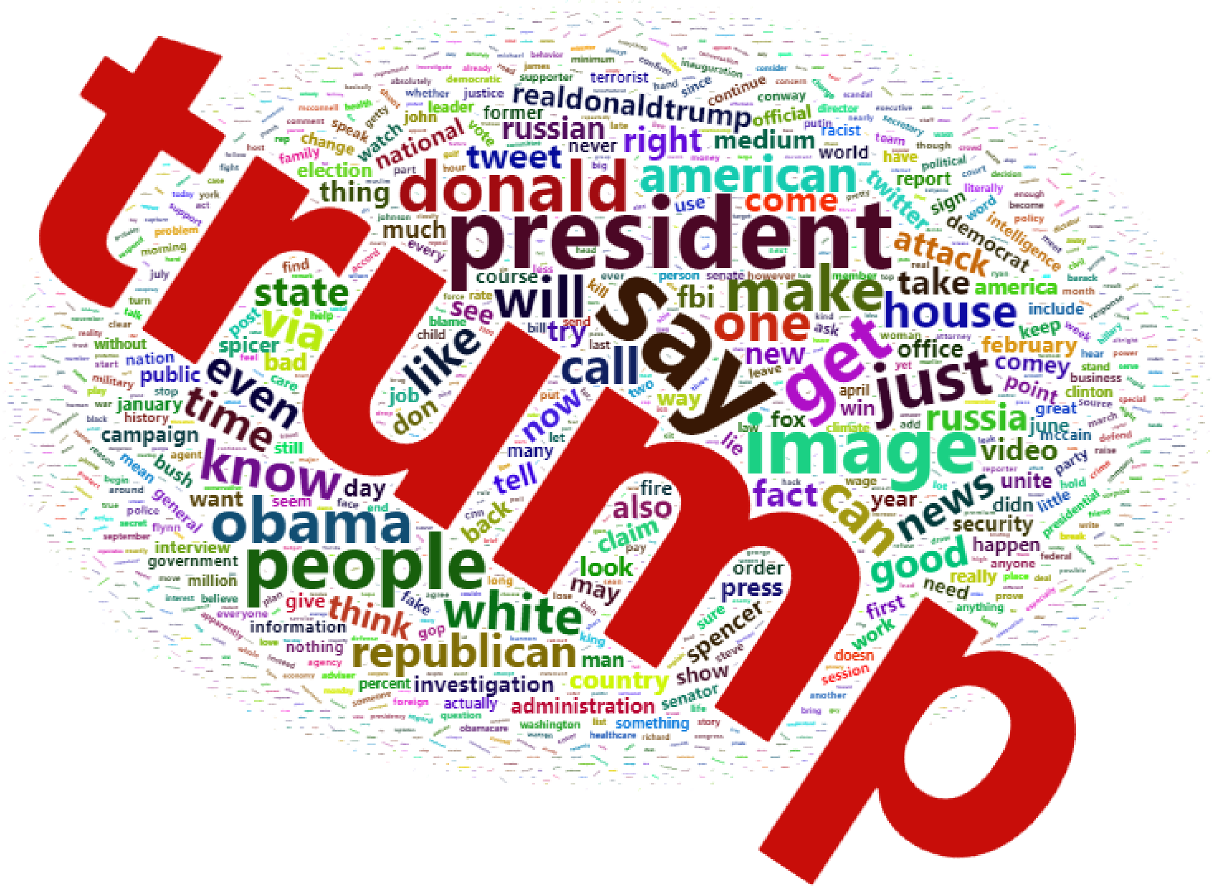
Figure 4.1 shows part of the data "wrangling"/data cleaning process in the R-Script. The point of data "wrangling", as mentioned in Introduction, is to clean up the human language. Natural Language Processing (NLP) means that we have to clean up the human language text. We use VCorpus function in R-Studio to clean up the dataset because it contains a lot of punctuation, symbols, and unwanted texts. For data cleaning, we convert all the texts to lower case, we remove the numbers, punctuations, stopwords, and whitespace. Lemmatization is the process of doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word.

For figure 4.2 and figure 4.3, we are removing all the terms whose sparsity is greater than a certain threshold value. A document term matrix is a mathematical matrix that describes the frequency of terms that occur in the dataset in a collection of documents. The rows correspond to documents in the collection and columns correspond to terms. Document term matrix is really useful in Natural Language

Processing. In our case, we can see that figure 4.3 shows after the removal of the sparsity terms, with sparse = 0.99.

In figure 4.2, we can see that on the first line there are 200 data entries, which has over 6792 that appeared at least once. We will remove any terms that do not appear in at least 1% (sparse = 0.99) of the data entries/documents. The Non-/sparse entries show there are 1,326,118 cells in frequencies that are zero, 32282 of the entries are non-zero figures. 98% of the cells are zero (1326118/(1326118+32282)). Figure 4.3, then, shows after removing the sparse terms, the number of documents is still 200, but the number of terms that have appeared at least once has decreased to 2444. The Non-sparse entries show that the total number of cells in frequencies that are zero decreases to 461,887, and the non-zero figures decrease to 26913. 94% of the cells are zero (461887/(461887+26913)).

Figure 5.1 Words Cloud for Fake News



Word Clouds, or wordle/word collage/tag cloud, are visual representations of words that give greater prominence to words that appear more frequently. In this case, we can see that the word “trump” stands out from the word cloud. One of the things that we have to remember is that it displays the frequencies of the words, not the importance. The larger the word is in the visualization the more common the word is in the Fake news documents. This means that out of all the fake news, “trump” is the most common word that appears in the Fake News. Other words that also appear really common are “say”, “president”, “image”, “donald”, “people”, “obama”, “make”, “house”, “get”, and etc.

Figure 5.2 Words Cloud for True News

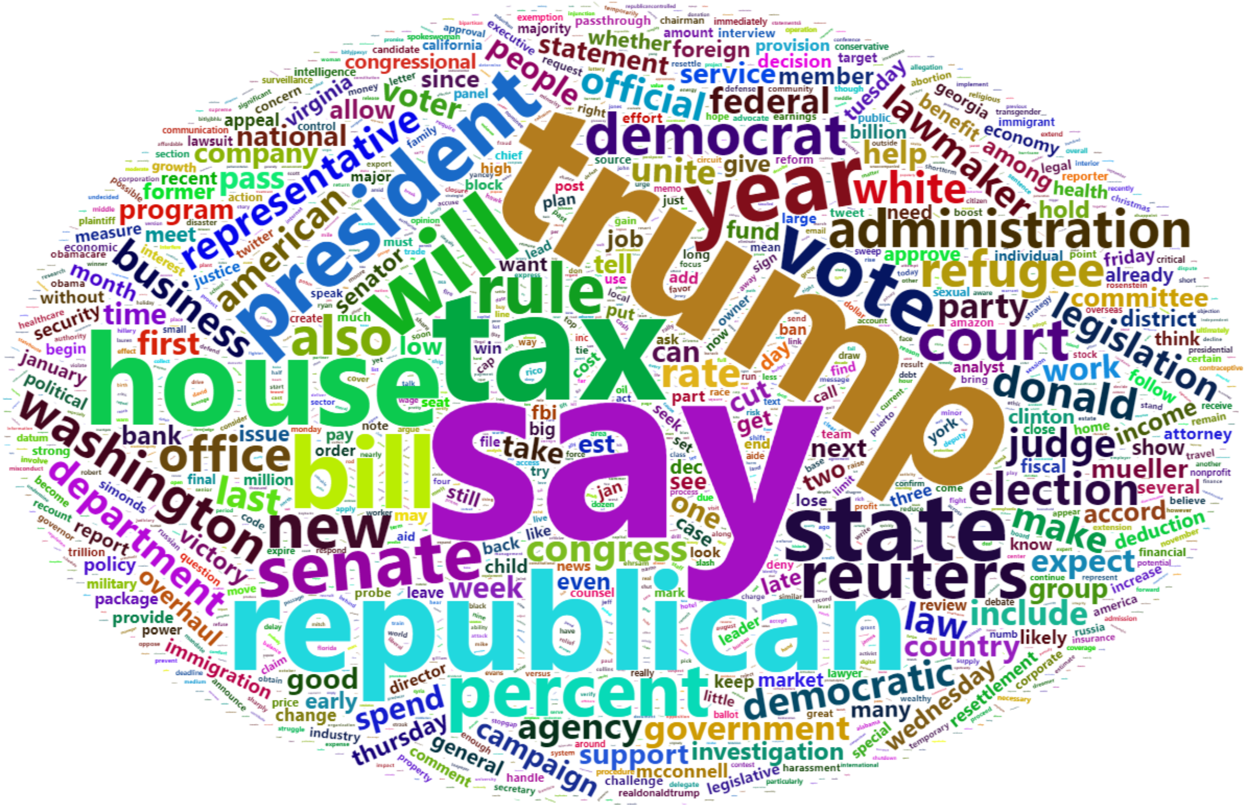
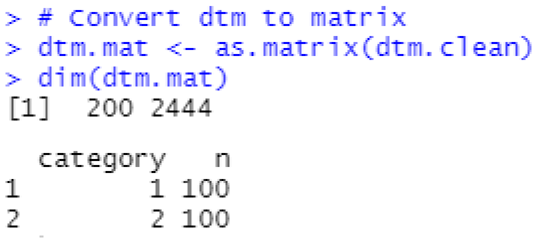


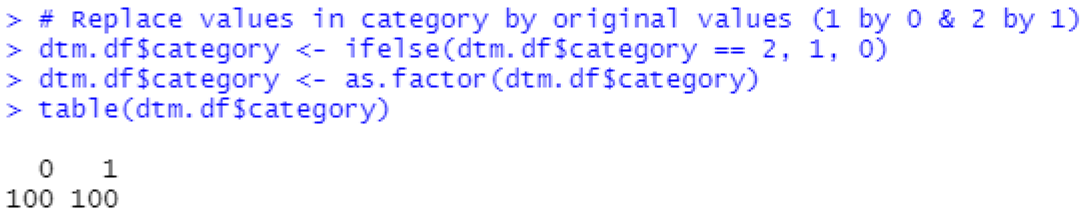
Figure 5 is a word cloud for True news. Out of all the true news, the most common word is “say”, followed by “trump”, “republican”, “tax”, “house”, “bill”, and “will”. It is shown as the larger the word in the visual the more common the word was in the documents of True news.

Figure 6.1, 6.2 & 6.3



```
> # Convert dtm to matrix
> dtm.mat <- as.matrix(dtm.clean)
> dim(dtm.mat)
[1]  200 2444

  category   n
1        1 100
2        2 100
```

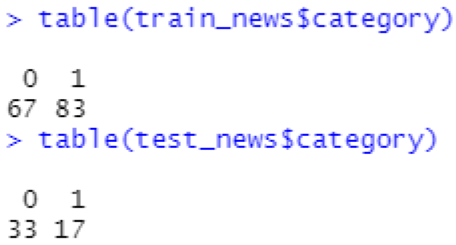


```
> # Replace values in category by original values (1 by 0 & 2 by 1)
> dtm.df$category <- ifelse(dtm.df$category == 2, 1, 0)
> dtm.df$category <- as.factor(dtm.df$category)
> table(dtm.df$category)

  0   1 
100 100 
```

Figure 6.1 shows that there are 2444 observations in total, and 200 random samples of observations that we have chosen during the pre-processing of data for further analysis. Figure 6.2 shows 0: Fake news and 1: True news have been transformed into 1: Fake news and 2: True news, with 100 observations of each. Figure 6.3 shows that we've replaced values in Category back with original values, with 1 by 0 for Fake news and 2 by 1 for True news, and there are 100 observations of each with 200 observations in total for the dataset.

Figure 7 Splitting Data into training & testing sets



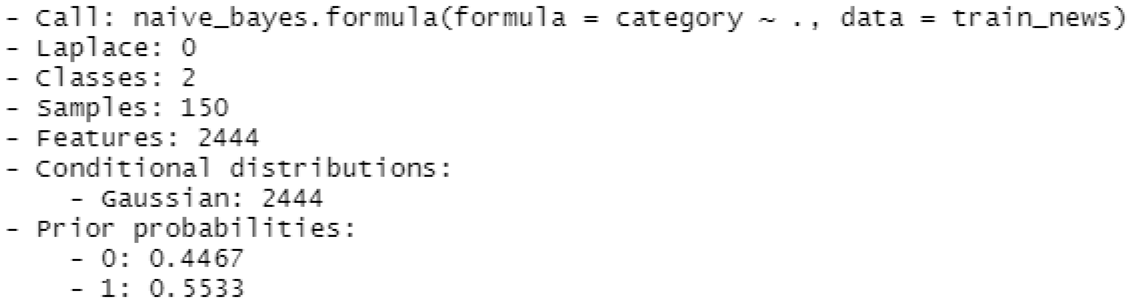
```
> table(train_news$category)

 0  1 
67 83 
> table(test_news$category)

 0  1 
33 17 
```

From Figure 7, we can see that the data is split into training and testing sets, using the 75/25 rule. The training set has a total of 150 observations, including 67 fake news observations and 83 real news observations. The test set has a total of 50 observations, including 33 fake news observations and 17 real news observations. The ratio of training set to testing set is 150:50, thus, 75:25. Therefore, we can confirm that the training and testing sets are balanced, and we can dive into classification methods like Naive Bayes Model and Logistic Regression.

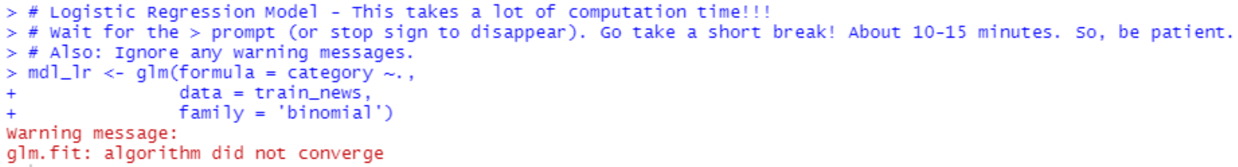
Figure 8 Naive Bayes Model



```
- Call: naive_bayes.formula(formula = category ~ ., data = train_news) 
- Laplace: 0 
- Classes: 2 
- Samples: 150 
- Features: 2444 
- Conditional distributions: 
    - Gaussian: 2444
- Prior probabilities: 
    - 0: 0.4467
    - 1: 0.5533
```

This figure shows the Naive Bayes Model of the dataset. The preprocessing process of data "wrangling", or data cleaning, consists of 5 variables of text data splitting into a training and a testing set, using the 75/25 rule, as mentioned in the Introduction. It involves removing numbers, punctuations, stop-words, stemming of words, and lemmatization of words. Laplace is a smoothing technique that helps tackle the problem of zero probability in the Naive Bayes Model. Our Laplace value is 0, meaning there is no smoothing happening. The Classes indicate that there are two classes (0: fake news, 1: real news) for the prediction/classification model. The Samples tell us that there are 150 observations in the training set used to compute the Naive Bayes Model, which are confirmed when the training and testing sets are split into 150:25, or 75:25, ratio. The Features show that there are 2444 text data observations after data cleaning. The prior probabilities show the likelihood of the new observations being fake news (0: 44.67%) or real news (1: 55.33%).

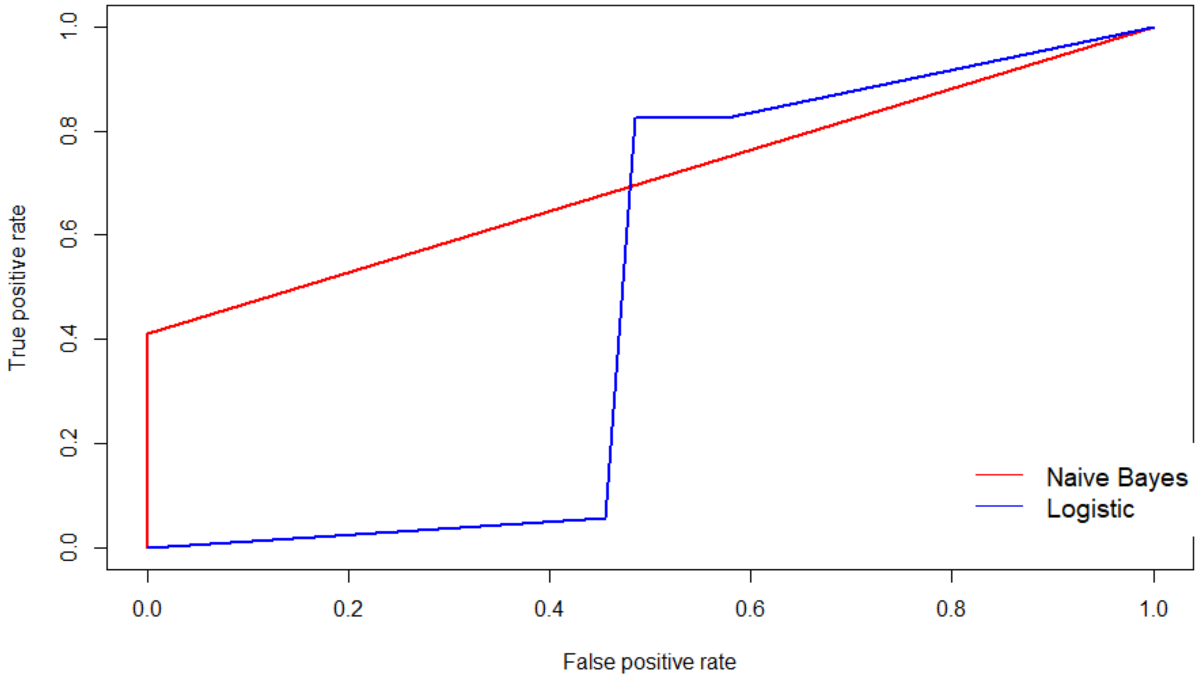
Figure 9 Logistic Regression Model



```
> # Logistic Regression Model - This takes a lot of computation time!!!
> # Wait for the > prompt (or stop sign to disappear). Go take a short break! About 10-15 minutes. So, be patient.
> # Also: Ignore any warning messages.
> mdl_lr <- glm(formula = category ~.,
+               data = train_news,
+               family = 'binomial')
Warning message:
glm.fit: algorithm did not converge
```

After running the R-script of the Logistic Regression Model, we anticipate getting a warning message. This is completely fine as the results will be shown in later visualizations.

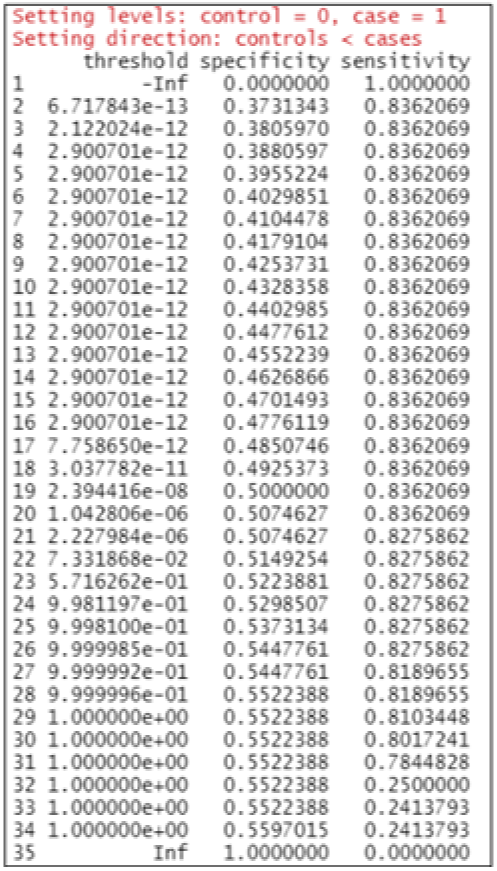
Figure 10 ROC Curve



The Receiver Operating Characteristics (ROC) curve, or the Lift curve, is a plot of true positive rate vs. false positive rate for different values of the cutoff probability for classification. It also shows the trade-off between sensitivity and specificity. Sensitivity is the true positive rate and specificity is 1-false positive rate. When we compare the different models of classification, we compare the Area Under the Curve (AUC). The curve resulting in a larger area under the ROC curve is favored over the one that yields a smaller area. The closer the curve is to the top-left corner, the more accurate it is at predicting the data. The greater the area indicates that the model is more effective under classification. The horizontal line at the top has 100% sensitivity and 100% specificity at every point, which is referred to as the perfect test. Each point on the curve corresponds to a different cutoff. So the way we construct the curve is by taking all the individual cut off points of a given test, and recalculating specificity and sensitivity for each cut off point and plot them.

Although we do not have the exact number for area under the curve, Naive Bayes ROC curve does look like it yields a larger area compared to the Logistic regression ROC curve, as it has almost double the area of the Logistic regression ROC curve. This means that, in this case, the Naive Bayes Model is more effective in terms of classifying the observations in the dataset of Real and Fake news.

Figure 11 Setting Threshold for Logistic Regression Model (Exhibit 27 given by Professor Mathaisel)

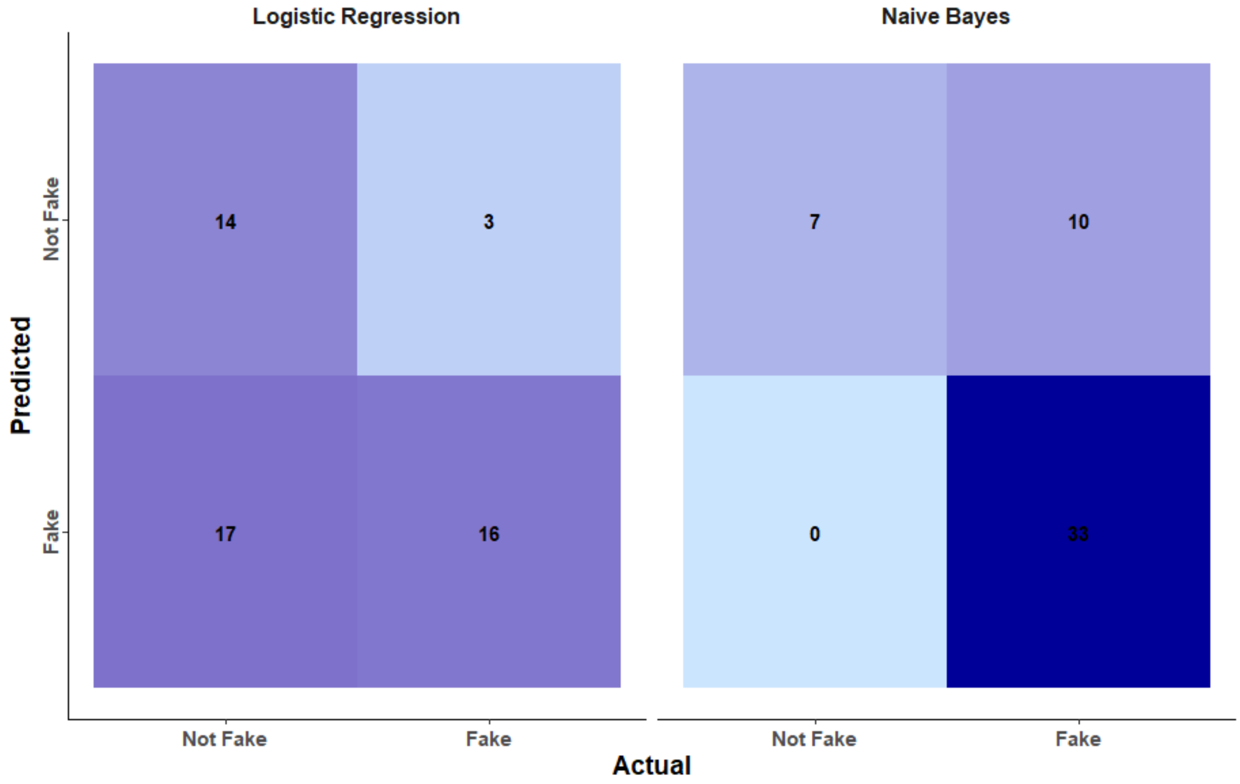


```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
      threshold specificity sensitivity
1          -Inf   0.0000000   1.0000000
2  6.717843e-13   0.3731343   0.8362069
3  2.122024e-12   0.3805970   0.8362069
4  2.900701e-12   0.3880597   0.8362069
5  2.900701e-12   0.3955224   0.8362069
6  2.900701e-12   0.4029851   0.8362069
7  2.900701e-12   0.4104478   0.8362069
8  2.900701e-12   0.4179104   0.8362069
9  2.900701e-12   0.4253731   0.8362069
10 2.900701e-12   0.4328358   0.8362069
11 2.900701e-12   0.4402985   0.8362069
12 2.900701e-12   0.4477612   0.8362069
13 2.900701e-12   0.4552239   0.8362069
14 2.900701e-12   0.4626866   0.8362069
15 2.900701e-12   0.4701493   0.8362069
16 2.900701e-12   0.4776119   0.8362069
17 7.758650e-12   0.4850746   0.8362069
18 3.037782e-11   0.4925373   0.8362069
19 2.394416e-08   0.5000000   0.8362069
20 1.042806e-06   0.5074627   0.8362069
21 2.227984e-06   0.5074627   0.8275862
22 7.331868e-02   0.5149254   0.8275862
23 5.716262e-01   0.5223881   0.8275862
24 9.981197e-01   0.5298507   0.8275862
25 9.998100e-01   0.5373134   0.8275862
26 9.999985e-01   0.5447761   0.8275862
27 9.999992e-01   0.5447761   0.8189655
28 9.999996e-01   0.5522388   0.8189655
29 1.000000e+00   0.5522388   0.8103448
30 1.000000e+00   0.5522388   0.8017241
31 1.000000e+00   0.5522388   0.7844828
32 1.000000e+00   0.5522388   0.2500000
33 1.000000e+00   0.5522388   0.2413793
34 1.000000e+00   0.5597015   0.2413793
35          Inf   1.0000000   0.0000000
```

Figure 11 shows the setted threshold for the logistic regression model. My Setting of threshold did not run in the R-script due to the package not installed properly through the internet. Therefore, I asked Professor Mathaisel to kindly share his output for this specific visualization. It is possible that my other visualizations did not have the similar result to this one because I used a random sample of 200 total observations, 100 from the Fake news dataset and 100 from True news dataset. Professor Mathaisel used a random sample of a total of 1000 observations, 500 from the Fake news dataset and 500 from True news dataset.

The output of a Logistic regression model is a probability. We can, therefore, select a threshold value. If the probability is greater than this threshold value, the event is predicted to happen otherwise it is predicted not to happen. A classification matrix, then compares the actual outcomes to the predicted outcomes.

Figure 12 Heat Map for Classification Matrix for Logistic Regression and Naive Bayes

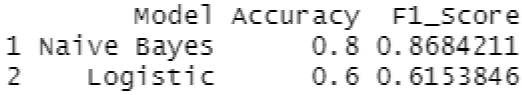


A heat map is a data visualization technique that shows the magnitude of a phenomenon as color in two dimensions. The variation in color may be by hue or intensity, giving obvious visual cues about how the classification prediction matches against what actually occurred.

For the Logistic Regression model Heat Map, it shows that the number of predicted observations that are Real news is 17, while the total number of predicted observations that are Fake news is 33. The actual number of Real news is 31 and the actual number of Fake news is 19. The number of Fake news that are correctly predicted to be Fake news is 16, and the number of Real news that are correctly predicted is 14. Therefore, the number of observations that are correctly predicted is (14+16=) 30. So the number of misclassified observations is 20. The misclassification rate, according to Logistic Regression model, is number of misclassified observations divided by the total number of observations being classified * 100%, which is (17+3=) 20/ (14+3+17+16=) 50 *100% = 40%.

For the Naive Bayes Model Heat Map, it shows that the number of predicted observations that are Real news is 17, and the number of predicted observations that are Fake news is 33. The actual number of Real news is 7, and the actual number of Fake news is 43. The number of Fake news that are correctly predicted to be Fake news is 33, and the number of Real news that are correctly predicted to be Real news is 7. Therefore, the number of observations that are correctly predicted is (33+7=) 40. So the number of misclassified observations is (0+10=) 10. The misclassification rate, according to the Naive Bayes model, is number of misclassified observations divided by the total number of observations being classified * 100%, which is (0+10=) 10/ (7+10+0+33=) 50 *100% = 20%.

Figure 13 Accuracy and F1 Score of Naive Bayes Model and Logistic Regression Model



```
        Model Accuracy  F1_Score
1 Naive Bayes      0.8 0.8684211
2    Logistic      0.6 0.6153846
```

Figure 13 tells us the accuracy of predicting the observations for Naive Bayes Model and Logistic Regression Model. The accuracy of the Naive Bayes Model to predict the new observations is 80% and the accuracy of the Logistic Regression Model to predict the new observations is 60%. Naive Bayes model has a higher accuracy compared to Logistic regression model.

The F1 score can be interpreted as a weighted average of precision and recall values, where an F1 score reaches its best value at 1 and worst value at 0. Precision is the number of true positives divided by the number of true positives and false positives. It is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV). Recall is the number of true positives divided by the number of true positives and the number of false negatives. It is the number of positive predictions divided by the number of positive class values in the test dta. It is also called Sensitivity or the True Positive Rate.

Looking at the F1 Score, we can see that the Logistic Regression model got a score of around 0.62 and Naive Bayes Model got a score of around 0.87. This means that Naive Bayes Model has a better prediction/classification in this Real and Fake news dataset compared to Logistic Regression model.

Conclusion

The aim of this executive summary is to build a logistic regression model and Naive Bayes model to classify and predict whether the new observations (news) is fake news or real news based on the characteristics of the newspaper. In Figure 1, it shows that there are no missing values in the 5 variables, “title”, “text”, “subject”, “date”, and “category”.

Figure 2 shows that there is a random sample of 100 observations in Fake news (0 for Category) and a random sample of 100 observations in Real news (1 for Category). Figure 3 shows that there is a random sample of 100 News and a random sample of 100 Politics News, for the Subject variable.

Figure 4.1 shows the data “wrangling” or data cleaning process in the R-Studio and the script. Since our data is in text data type, we have to clean up the human language text (Natural Language Processing). We convert all the texts to lower case, remove the numbers, puncturations, stopwords, and whitespace. We also stemmed and lemmatized the words and removed inflectional endings only to return the base or dictionary form of a word. Figure 4.2 and 4.3 shows the process of removing all the terms whose sparsity is greater than a certain threshold value, which in our case is sparse = 0.99. The number of terms that have appeared at least once has decreased to 2444. The Non-sparse entries show that the total number of cells in frequencies that are zero decreases from 1326118 to 461887, and the non-zero figures decrease from 32282 to 26913.

Figure 5.1 and figure 5.2 are word clouds for Fake news and True news, respectively. The most common words in the Fake news observations are “trump”, followed by words like “say”, “president”, “image”, “donald”, “people”, “obama”, “make”, “house”, “get”, and etc. The most common words in the Real news observations is “say”, followed by words like “trump”, “republican”, “tax”, “house”, “bill”, and “will”.

Figure 6.1 shows that there are 2444 observations in total, after the data cleaning process, and a random sample of 200 observations from Fake dataset and True dataset. Figure 6.2 shows that the Category is replaced with 0 by 1 for Fake news and 1 by 2 for True news. Figure 6.3 shows that we’ve replaced values in Category back with the original values of 1 by 0 for Fake news and 2 by 1 for True news.

Figure 7 shows that the dataset is split into a training set and a testing set with 150 observations in the training set and 50 observations in the testing set, which match up with the method of 75/25 rule with a ratio of 150:50 = 75:25. This is very important before we dive into our classification methods of Naive Bayes model and Logistic Regression model.

Figure 8 shows the summary of Naive Bayes model of the dataset. The prior probabilities show the likelihood of the new observation being Fake news is 44.67% and the likelihood of the new observation being Real news is 55.33%. Figure 9 shows the running of the R-script for the Logistic Regression model. We anticipated a warning message, as shown in the figure. The result of the model will be analyzed later in the Conclusion. Figure 11 shows the process of setting threshold for Logistic Regression Model. The output of a Logistic Regression model is a probability. If the probability is greater than this threshold value, the event is predicted to happen otherwise it is predicted not to happen.

Figure 10 is the ROC curve of both Naive Bayes model and Logistic Regression model. As we’ve mentioned in the analysis, we do not have an exact number for the area under the curve (AUC) for both Naive Bayes model curve and Logistic regression model curve, but we can eyeball it and tell that Naive Bayes model ROC curve yields a larger area compared to the Logistic regression model ROC curve, by almost double the area. This means that the Naive Bayes model is more effective in terms of classifying and predicting the observations in the dataset of Real and Fake news. Figure 12 is a Heat Map for Classification Matrix for Logistic Regression model and Naive Bayes model. It displays a similar result to figure 10. The Logistic Regression model has a misclassification rate of 40% and the Naive Bayes model has a misclassification rate of 20%. Therefore, the Naive Bayes model has a lower misclassification rate, which means that it can predict more accurately compared to the Logistic regression model. Figure 13 also yields a similar result to Figure 10 and Figure 12. Naive Bayes model has a higher accuracy of 80% compared to Logistic regression model's accuracy of 60%. Naive Bayes model also has a higher F1 score of approximately 0.87 compared to Logistic regression model's F1 score of approximately 0.62.

In conclusion, the Naive Bayes model has a higher predictive and classification power and accuracy compared to the Logistic regression model for this dataset of Real or Fake news.