FAKE NEWS CLASSIFICAION

Data Analytics

A PROJECT REPORT

Submitted by

Kathal Aditya Rajendra 18BCD7008

Sharaj Raja Chandran 18BCD7044

Under the Guidance of

Dr. Awnish Kumar

Dr. Sumathi D.

Associate Professor, CSE,

VIT-AP

TABLE OF CONTENTS

Chapter -1

Introduction

In today's modern era, information (both credible and otherwise) is available within a number of clicks. As a result, fake news has become a new rising threat whose rippling effect can be seen throughout the world. In the words of Churchill: "A lie gets halfway around the world before truth puts on its boots." Word travels quick and rather than knowing its source people are naturally more concerned about its content and this ignorance leads to misinformation being spread.

This has been very apparent in the most recent of times due to the heavy political drama taking place in many of the 1st World nations most of which can be attributed to counterfeit news spreading like wildfire. Misinformation can lead to devastating outcomes which overshadow real problems of the world and give rise to a greater number of problems which have been purposefully created. As a result, it has become a necessity to vet information thoroughly so that spread of misinformation can be curbed.

To check if a certain piece of information is real or fake is a very tedious task and people rather just accept it and underplay its grave consequences. In order to verify any news, the information must be gathered from multiple trustworthy sources. Getting the information first hand is the best method and even then, one could be getting tricked by lies. To overcome all their problems, it is better to develop a method which classifies data as real or fake quickly by observing past media which could at least warn people when a news snippet may be fake.

Through this project we aim to classify news as Real and Fake using Text Classification techniques which are described in the paper by Hadeer Ahmed, Issa Traore and Sherif Saad published in Wiley Journal in the year 2017.

We will be using the concepts which we learned during the Data Analytics (CSE4027) course such as Data Visualization, Data Cleaning, Statistical techniques and various other R functions in order to prepare the data for further Text Classification operations. By doing so we will be applying the concepts taught in class and then some which will ensure a good quality. The very same model which is used to classify news as genuine or fake can also be implemented in order to categorize SMSs as legitimate and spam.

Chapter -2

Background Study

Whilst researching for this topic we found that a number of research papers have been written on the topic of "Text Based Classification" and throughout the years the methodology applied has been refined to yield better results. The premise is simple to understand; vast amount of already classified text is taken as input for training the model so that future data can be predicted as to fall under which category.

The concept of text classification has also started to shift from Data Analysis and Data Mining to Machine Learning. The idea is to create completely automated systems which could reliably tag data under the categories of real and fake. Though this approach is it its early stages and text classification though Data analysis and mining algorithms have been well established.

Though the format of the input text may vary and the terms of classification may change, the algorithms used still remain mostly the same. Depending on the format of the said input texts different preprocessing techniques are applied to clean the data so that the classification algorithms can be applied properly and provide better results.

We have taken inspiration from one such paper; Wiley Journal (2017) - Detecting opinion spams and fake news using text classification by Hadeer Ahmed, Issa Traore and Sherif Saad. In this paper they apply various classification techniques individually on the data in order to determine which technique gives the most optimal result. They studied 6 different classifiers to predict the class of the documents, including SGD, SVM, LSVM, LR, KNN, and DT.

The various models are applied on processed data individually and the final results are then compared to find which models work better. But it was clearly concluded by them that different models turn out to give the best results depending on the kind of data that is fed to the program.

We intend to use their work to further build a model which is more accurate in nature and works on many more kinds of textual inputs. Apart from applying the different algorithms individually, we plan to use ensemble methods in order to use multiple models in conjunction so that the results yielded are much more accurate and the program as a whole works better for a large variety of data inputs.

Chapter -3

Problem Definition

The world is overflowing with data which is also available to us readily thanks to the massive amounts of electronic communication gadgets which surround us in our day to day lives. As a result, we are subjected to a tonne of information and a significant part of it is in the form of news.

News media have a great responsibility on them to only project accurate and trustable information but sadly this has not been the case in recent times. News is warped and manipulated in order to spread personal biases and affect the impressionable audience which gets subjected to it. On top of that news which travel through social media tends to be more dangerous as it has no trustable sources or anyone who can be held accountable for it.

As a result of all this it is necessary to develop a method by which information can be vetted accurately and swiftly in order to keep up with all news flowing through the world. This task is almost impossible to be done manually and hence requires the help of programmed models which can classify news as real and fake.

The data we receive will be in the form of text. Additionally, it will also have attached with it a date of when the news was published and the main topic of the news. Though the headline and the main body of the content is what is off importance to us. The model to be developed will deal heavily with strings and characters rather than numerical inputs which are basically non-existent.

The final output of this project should be a dataset of processed data which has been classified distinctly into two categories. Any piece of information which is regarded as news can only be completely true or false. Even the smallest amount of fallacy in the text of the news makes it null and void and then can be tagged as fake. The real news will be represented by the character ‘1’ whereas the fake news will be characterized as ‘0’.

Chapter -4

Objective

The main objective of this project is to classify the inputted news text as real and fake as accurately as possible. Any number of news fed as input will be tagged as ‘1’ or ‘0’ depending on whether it is real or fake respectively.

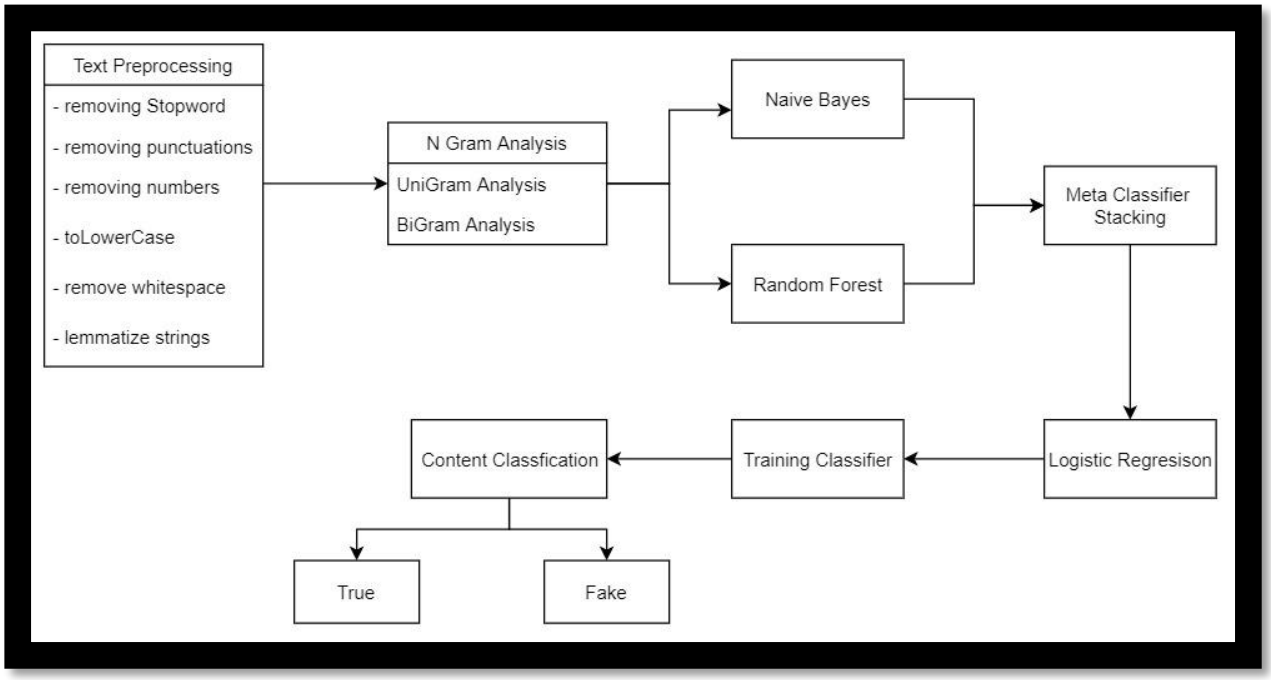
Though this goal cannot be achieved directly and with such ease as the above statement may imply. There are a number of smaller objectives that must be reached and surpassed so that our model may yield the most accurate result as possible. The objectives which we took into consideration are:

- The data first needs to be cleaned so that only the essential information is retained which will give accurate results.
- Then the format of said data is changed so that it may better fit the model.
- The data is then visualized so that a better understanding is developed about it and we can judge what kind of models are to be applied.
- Next the models are applied on the data, at first individually and then in conjunction
- At last, the results we get from all the various different models are compared and the necessary conclusions are drawn.

By completing these smaller objectives, the actual goal of the project will be achieved. The news text which gets fed into the program will be classified and tagged as real or fake with high accuracy.

Chapter -5

Methodology/Procedure



The above pasted diagram represents the procedure and flow of the program which we developed for our project. It is very easy to understand as the flow of the program is quite linear and it only contain one fork overall which represent the two techniques which have been applied on the processed data.

The first step as is with any data-based project is that of preprocessing. This involves either the removal or replacement of all the null and void entries of the data set. In our case we found that the number of null values was very low and hence just dropping the necessary columns would negligibly affect the overall result. Then we remove all the unwanted information that is included in the dataset. This includes all the stop words which are the words which occur very frequently but have very less to none affect on the overall data such as the, and, as, etc. Next the numerical values are deleted as they serve no purpose to this project and its result. Punctuations and other special characters are removed for the same reason. Next the data is formatted in such a way that it can easily be applied to the models. All letters are changed to lower case as the case of the letters is inconsequential. All blank spaces are removed as the program can easily group letters in such a way that the form words. Further more the data is lemmatized i.e., words representing the same base word or meaning are considered the same base word. (Ex. Running, runner, runs all are considered as just run)

The next step after preprocessing is to do the N-Gram Analysis of the dataset. We have in particular used Uni-Gram and Bi-Gram Analysis in our project. In Uni-Gram analysis the frequency of the individual terms is found and represented in the form of a graph. Though this

can be also done by using the simple inbuilt commands. We do this so that a comparison can be drawn with the results of the Bi-Gram Analysis where words which appear together are considered one unit and their frequencies are generated and represented using another graph.

Next, we apply two different classification models, namely Random Forest Classification and Naïve Bayes Classification. It is important to note that these models are applied individually on the dataset. The data gets split into a training and testing section in order to prevent leaking. We use the training subset of the data in order to train the models and use the testing subset in order to compare the predicted values with the actual values. The accuracy is calculated for the prediction values from both of these classification techniques.

Algorithm 1 - Stacking

Input : $D = \{(x_i, y_i) | x_i \in \chi,\ y_i \in Y\}$

Output : An ensemble classifier H

1. **Step 1 :** Learn first-level classifiers
2. For $t \leftarrow 1$ to T do
3. Learn a base classifier h_t based on D
4. **Step 2 :** Construct new data set from D
5. For $i \leftarrow 1$ *to* m do
6. Construct a new data set that contains $\{x_i^{new}, y_i\}$, where $x_i^{new} = \{h_j(x_i)\ for\ j = 1\ to\ T\}$
7. **Step 3 :** Learn a second-level classifier
8. Learn a new classifier h^{new} based on the newly constructed data set
9. **Return** $H(x) = h^{new}(h_1(x), h_2(x), \ldots, h_T(x))$

Following the application of these models, we apply the concept of stacking using meta classifiers. Stacking involves combining multiple classification techniques using meta classifiers in order to get highly predictive values. The predictive features of the base level classifiers are used as input for the meta classifier.

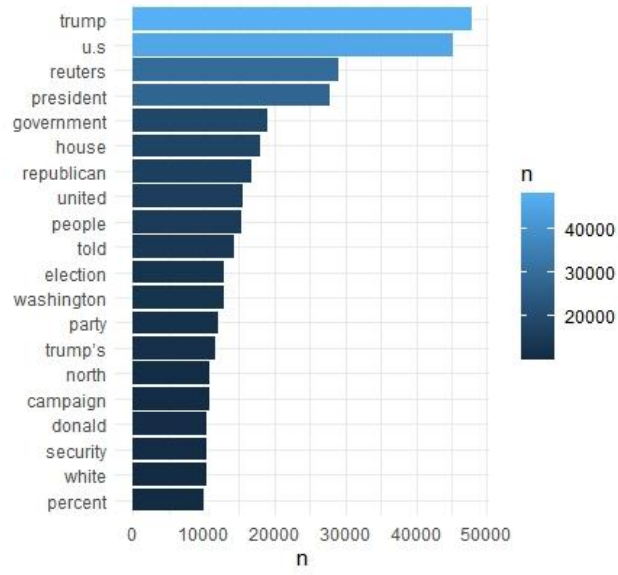
The output is binary in nature as the classification is done as real or fake and represented by ‘1’ and ‘0’ respectively. In our case as we used meta classifier hence we apply logistic regression. Logistic regression is used for binary classification and is used in our project as an ensemble classifier.

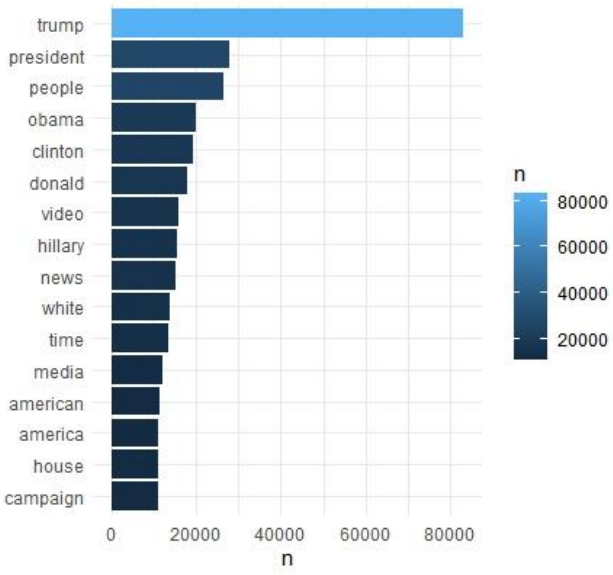
Finally, the data is classified into two categories namely true and sake which is represented by the numbers 1 and 0 respectively. The final visualization of the project is done through a number of confusion matrices which make it easy to compare and contrast the classification of data which has been done using the various models.

Chapter -6

Results and Discussion

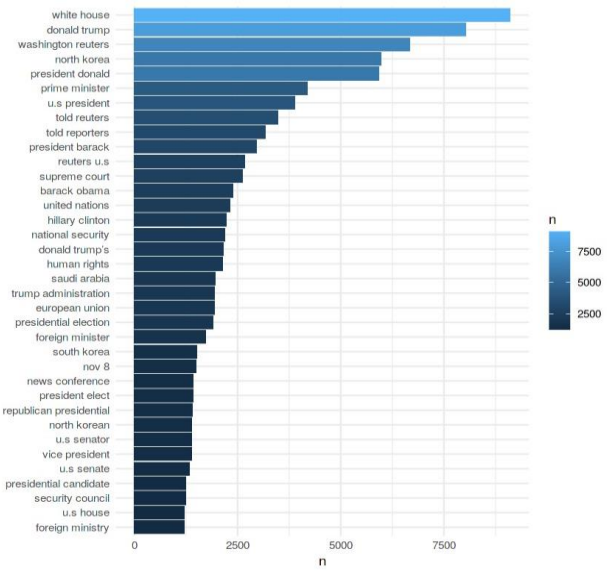
UNI-GRAM ANALYSIS

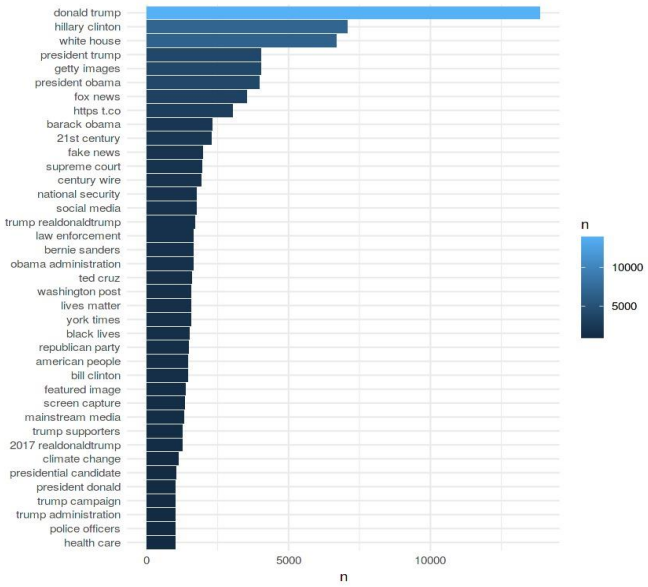




REAL FAKE

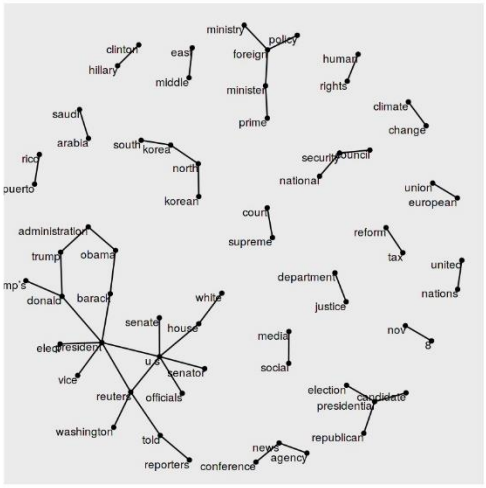
The Uni-Gram analysis of the true and false datasets clearly showed that the word “u.s” appeared significantly greater number of times than in the fake dataset and hence is a clear telltale sign in order to classify the news as real and fake. Though it is also clear that many words have appeared with very high and similar frequencies in both the graphs such as “trump”, “president”, “people”, etc. Hence these words need to be avoided as factors involved in classification as they are present in both fake as well as real news in high numbers.

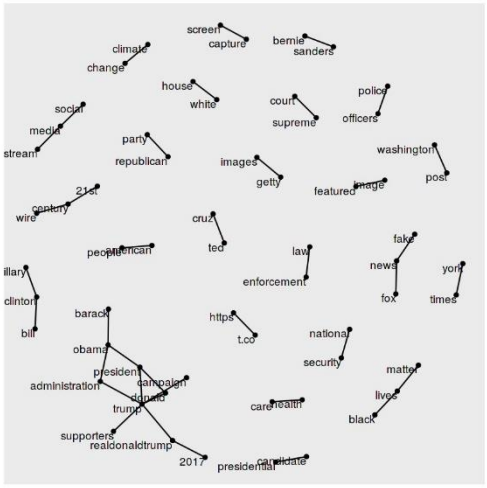




REAL FAKE

The graphs above are the visual representation of the data after the Bi-Gram Analysis. As you can see the words in question have clearly been grouped in sets of two as explained before. The words “donald trump” have appeared in a significantly higher number in the Fake news rather than the real news. Also, the term “White house” appears seems to appear much more in legitimate pieces of news media. Both of these N-Gram Analysis techniques help us find the words we need to look out for in order to distinguish between real and fake information.

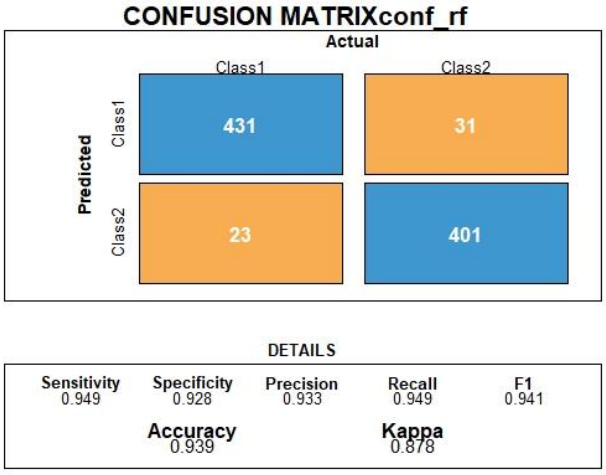


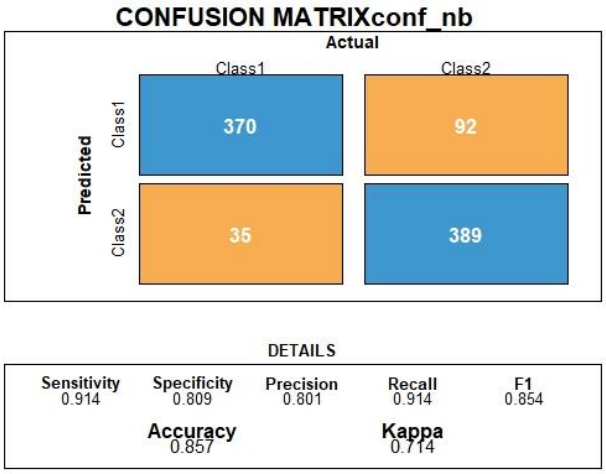


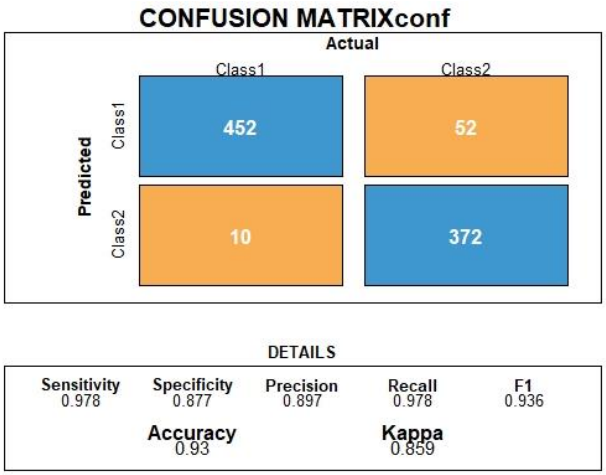
REAL FAKE

The above graphs represent the association between words which we found. The first graph shows the associations of words which appear in the texts of Real News. A big cluster is clearly visible which tells us that the words present in the said clusters are used more often together in real news. In contradiction, the graph representing the association of fake news words has a relatively smaller cluster with a few similar words. We can compare these clusters and easily understand which combination of words appear in what kind (real/fake) of news media. One such example is that “2017” is present in the fake news cluster and not the real news clusters which clearly hints that news coming out about past years may more likely be fake.

CONFUSION MATRICES







It is clearly visible that the accuracy of Naïve Based Classifier is the worst. The Random Forest on the other hand yields the best accuracy slightly surpassing the meta classifier. The main point to be noted here is that the best recall value is generated by the meta classifier. Hence, we can say that even though the base classifiers may give better accuracy than the meta classifier, the meta classifiers is still the better fit for the model. Additionally, the performance of the base classifiers may change from one dataset to another whereas the meta classifier used by stacking will always perform better as a whole under general circumstances.

Chapter -7

Conclusion and Future Scope

After looking at the results we get by applying the different models individually it is clear that Random Forest Classifier works better than Naïve Bayes Classifier and also does slightly better than logistic regression which is used as a meta classifier for stacking.

The Meta Classifier has the highest recall value out of all the classifiers used and hence is the best model applied to the dataset. It will perform best in general as it will use the combination of the base classifiers in order it order to yield highly predictable values.

The Ensemble Techniques yield a high recall value which suggests that the classification techniques which have been used are very good fits for the data and provide relevant classification and results.

The same model can be applied for any binary classification application which has its input as text. Though minor changes may be required but most of the code overall will still remain the same.

Due to the use of a meta classifier, the project can be applied on a wide range of datasets in order to perform binary classification. Where some base classifiers may yield less accurate results the meta classifier will give a more accurate prediction.

With higher computational power if we aim to apply more of such robust models then we can greatly reduce the amount of misinformation which is floating in the world in the form of fake news which will in turn inhibit the spread of mass hysteria and let people focus on the real issues at hand. It will overall cause greater development to take place in the world and solve a lot of the underlying issues which curb the progress of the nations.

References (Sample)

- https://www.oreilly.com/library/view/text-mining-with/9781491981641/ch01.html
- https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.9
- https://www.kdnuggets.com/2017/09/ensemble-learning-improve-machine-learningresults.html/2
- https://dzone.com/articles/ensemble-learning-to-improve-machine-learningresu
- https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemblemodels/
- https://web.njit.edu/~avp38/projects/multi_projects/ensemble.html
- https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php

Appendix – A

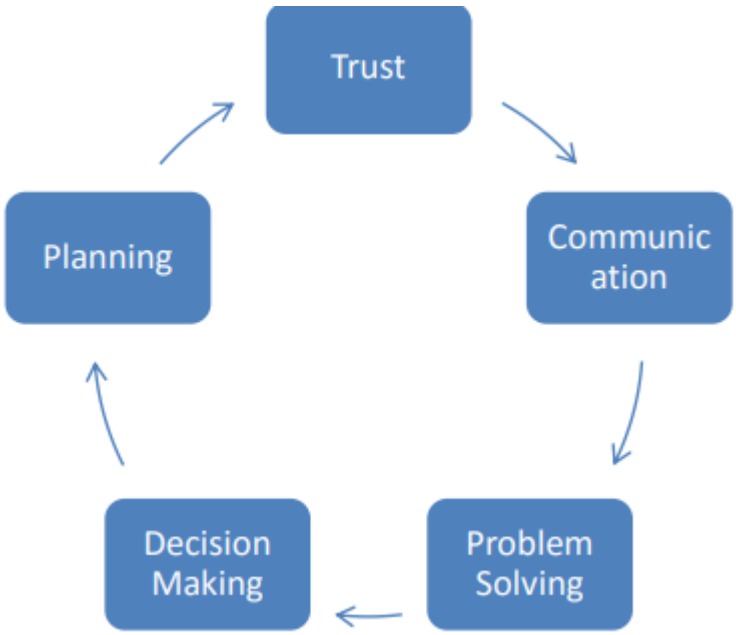
Team Work and Work Management

Theoretical Concepts used:

Team Building:

For this project the format thing we were needed to form a team of two. As taught, we formed teams to complement each other's weakness. Sharaj is better at documentation and research required for any topic. He researchers a topic in every possible aspect and provide the results of this through research to his team mates. While Aditya is good at coding and implementing theoretical model into code. So, we had clear goal setting and work one need to do.

When we look from the 5 steps involved in the team building:



We start our discussion from the trust component of this cycle. We both trust each other's ability to complete the task assigned and know our own weakness so we divided the tasks such that each one is working on a task he can complete within the given deadline. Moving on we will discuss below how we established proper communication channels so that every information is received to each one of us.

For the next three components of the used the communication channels to share our ideas with each other and also solve problems has they arise. For planning we has extensive meeting discussing the project and how to implement it and research needed.

The decision system setup consists of each team member contacting the other and then discussion is held on Google meet whenever needed like presenting screen which cannot be done on voice call.

Going through the list that we studied during the course and answering how we used them to create a system so that each team member contributed to the project and everyone supported each other ideas while pursing the freedom in their own tasks:

- Clear Expectations

We both agreed that we need to make a project that helps us study new topics and apply a combination of past knowledge and topics learned during the course to make a deployable project. So, we had clear vision in our minds while choosing the project topic.

- Context – Background

Working in teams helped us in achieving the target in a timely manner with each member helping the other where he lacked in knowledge or experience. So, working in team help us doing the project in a phased manner and set SMART goals.

- Charter – agreement – Assigned area of responsibility

We have our roles defined very clearly and had clear division of work. We have already discussed this division of work above.

- Control – Freedom & Limitations

We had agreed that each member has full freedom to use any method and techniques to complete an objective until it is achieved with in the deadline and provided output as required. The only constraint we put in place was more of a check system, whenever we made any changes to the code one has to leave a comment explaining what he changed and sending a WhatsApp message with the similar message.

- Collaboration

During this COVID-19 situation collaboration was based done on online since no physical meeting could be done. We created a shared google colab notebook that so that each one can write our ideas as code simultaneously.

- Communication

For communication we used WhatsApp for just information sharing and when we something was needed to discussed on detail, we used Google meet to hold meeting. Proper communication channels were established with more emphasis on the written communication. So that we have records of the things we discussed and out plan forward. Verbal communication was used to share ideas during meetings while the WhatsApp messages and comments in code helped in the written communication.

- Coordination

We were able to properly coordinate tasks because of proper commutation channels and clear division of work. We knew what the other person was doing and what percentage of the objective he has achieved. This helped us in managing our own timelines so that we completed our objectives at the same time the other teammate completed while abiding with the common deadline.

Goal Setting:

As we learned during the course that SMART technique is the best for setting goals. We have also used the same technique while setting goals for our project. Now we will discuss each component of the SMART singularly.

S: Specific

Work activities should be specific. This is very important part of goal setting process. If the team members do not have their goals defined properly then they might do unnecessary or redundant work thus decreasing their productivity. So, when we were setting our timeline for the project and goals to be meets for the project completion, we made goals very specific to the strength of the team mate.

M: Measurable

The output metrics and yardsticks should be defined. If we do not know how much percentage of the work we have completed and with a unit a time, we cannot increase your speed or decrease when needed to match the final deadline. So, the goals we set were measurable and quantifiable. For example, one the goal set was to complete the graph and save them .PNG format before Sunday. So, we have 10 graphs and now we can calculate the percentage of work completed and how much more time is required which can be communicated to the other team member.

A: Achievable

Should be assigned to those responsible for achieving it. As we have already discussed upon this point that we had divided our work in such a work that it fits for the most part of it with our strengths, so that we can achieve our specific goal in a timely manner.

R: Realistic

Should be challenging yet attainable. Have a motivational effect. When we were selecting the project, we wanted it to be challenging so that we can learn something new and apply it on a real-world problem. So, we did not choose a non-solvable problem such of stock market prediction due to its volatile nature. We choose a project which we can complete with the given time and give our best effort into. We learned about ensemble techniques and their application.

T: Time bound

Time period for achievement is clearly stated. As this project also had an external deadline associated with it, we had to do out time management such that we complete everting by that deadline. The time management principles learned during the course were applied to manage time and completed goals.

Time Management Principles followed:

1. Planning: Planning is an integral part of executing a project. To have a well-established plan ensures that at no time are the group members lost on what needs to be done. Additionally, it is easy to understand on how far are they in the project and whether

they are on schedule or not. At the start of the project a detailed plan was drawn out which was followed throughout.

2. Organize and Prioritize:
    a. Eishenhower's Principle: All the urgent work was given the highest priority. Both the team members worked on the same task together so that the task can be completed swiftly. After which the important tasks were assigned to the individual members to be done properly and in good time.
3. The 80/20 Rule: Following the 80/20 rule, maximum effort was put into the most important of tasks such as data modeling and research which yielded the most important of results.
4. Do One Thing At A Time: We kept multi-tasking to a minimum and only assigned one task at a time to both the team members so as to not burden them as well as ensure that they put all their focus and attention on that one task so that it can be done well.
5. Avoid Distractions: During the time on which the team members worked on the project, all other unrelated tasks were kept on a halt. Additionally, they did not interact with any other electronic devices and only communicated with each other.
6. Delegate: As most of the time the tasks assigned to the two members were different, so they got completed in different amount of time. Hence whenever someone would complete their task, they would either go on to help the other or else take on another task. This ensured that continuous work was being done on the project as well as that time was not being wasted.

Actual Flow of the Project Work:

The team consisted of two members Aditya Rajendra Kathal (18BCD7008) and Sharaj Raja Chandran (18BCD7044), which made division of work easier. Both the members of the teams had very similar ability and expertise in the subject so working together was made much easier. Though the downside of this was that both the members lacked similar knowledge which was needed for certain parts of this project. Also, the team members had their own preferences on which kind of tasks they were comfortable and confident with. Thus, the distribution of work was done with all this in mind.

Furthermore, the two team members have worked together on a number of projects before. These projects have varied in nature from technical projects related to mathematics and computer science to hosting technical events and seminars. Due to all of this experience of working together for the past few years, they make a good team and work especially well together.

The work to be done in this particular project could be distinctly categorized as technical and non-technical. The non-technical work included everything from researching about the project, preparing the first abstract, preparing the necessary documents such as the Power Point Presentation, Final Report, etc. The technical tasks were everything in relation with the actual programming that was involved in this project. It consisted of jobs such as cleaning the data, applying the necessary algorithms, visualizing the result, etc.

As a result, we could divide the tasks in such a way that none of the two members was burdened with a purely technical or non-technical set of tasks. This is because it is important that there is variation in the kind of work assigned so that monotonicity can be avoided which discourages people from working due to boredom born out of the repetitive cycle of work which is assigned to them. Hence it was ensured that both the members at the least got assigned one of the technical as well as non-technical tasks.

When the work was started initially there was a schedule which was created. This schedule contained the tasks which were to be completed along with their deadlines. Rather than having one big deadline for the entire project, the project was broken down into simpler parts so that the amount of work that needed to be done could be evenly spread out in the time that was remaining. Also, it made such a big project look achievable even with a tight schedule and other additional works unrelated to the project.

The work began with researching for the project. Things such as finding existing paper, looking at concepts which could be applied, searching for a satisfactory dataset to work on etc. This task was done simultaneously by both the members individually. On the given deadlines both the members presented their findings and brainstormed in order to decide which project topic to work on. After reaching a unanimous decision the work started getting divided. Aditya was assigned to prepare the initial draft of the abstract which was needed to be submitted to the

faculty members. Simultaneously, Sharaj started work on the cleaning of the data which was required for any data heavy project.

Till now all members of the team were well experienced in the assignments they had been given as they have done the same tasks time and time again over the past few years on different projects but now, they were in uncharted waters. In order to progress further they needed a learn a few new concepts and theory which put the project work on sort of a hold. After they were confident that they had a good enough grasp on the new knowledge which they had attained, they resumed the work on the project.

The members switched the kind of tasks they were doing; Aditya was assigned the work of formatting as well as starting to apply the analytical and classification-based models on the data whereas Sharaj started work on the report which was to be submitted at the end of the project period. There was an open line of communication maintained throughout the ordeal as this was a necessity. There were changes made on both ends and it was vital that the changes be reflected at the other end to. Hence proper and efficient communication was important.

In order to ensure that the quality of our work was maintained throughout we checked each other's work and discussed thoroughly about each and every point so that both of us would be completely capable of answering any questions in relation with our project. Additionally, it also acted as a break from the regular tasks which we had assigned to one another.

Naturally the technical part of the project got over before the non-technical part as the final documentation of the project always takes place at the end. At that point of time upwards of eighty five percent of our work was done. We went through the whole program which we had developed multiple times and even tested it rigorously by inputting a variety of datasets which differed in dimensions, formatting, etc. After ensuring the competency of our code the next step was to complete the final report.

The technical task mostly had been done by Aditya, especially those which were remaining at the end. So, it was Sharaj's responsibility to complete the final report while Aditya worked on the aesthetics of the visualization code as well as went through everything which they had completed for the project so far. Aditya supplied the screenshots of the output which was integrated into the report by Sharaj who also completed the final report by documenting down all the necessary conclusions they had drawn from the project.

In conclusion, the whole experience of working on this project went very well. This was in parts both because of the efficient management of work which we had established as well as the team work which we had gotten accustomed to due to the numerous projects we had worked on before. The last few days of the project were a bit hectic but we pulled it off due to our trust in each other and proper work ethic which we had developed.

Appendix – B

Coding and Snap Shot

Coding:

Importing all the required modules

```
install.packages("readr")
install.packages("dplyr")
install.packages("stringr")
install.packages("ggplot2")
install.packages("tidyr")
install.packages("tm")
install.packages("textstem")
install.packages("tidytext")
install.packages("pROC")
install.packages("ROCR")
install.packages("randomForest")
install.packages("naivebayes")
install.packages("caret")
install.packages("janeaustenr")
install.packages("igraph")
install.packages("ggraph")
install.packages("e1071")
```

```
library(readr)
library(dplyr)
library(stringr)
library(ggplot2)
library(tidyr)
library(tm)
library(textstem)
library(tidytext)
library(pROC)
library(ROCR)
library(randomForest)
library(naivebayes)
library(caret)
library(janeaustenr)
library(igraph)
library(ggraph)
library(e1071)
```

```
fake <- read_csv('Fake.csv')
true <- read_csv('True.csv')
```

```
── Column specification ────────────────────────────────────────
cols(
  title = col_character(),
  text = col_character(),
  subject = col_character(),
  date = col_character()
)


── Column specification ────────────────────────────────────────
cols(
  title = col_character(),
  text = col_character(),
  subject = col_character(),
  date = col_character()
)
```

```
head(fake,1)
head(true,1)
```

A tibble: 1 × 4

title <chr>	text <chr>	subject <chr>	date <chr>
Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had to give a shout out to his enemies, haters and the very dishonest fake news media. The former reality show star had just one job to do and he couldn t do it. As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year, President Angry Pants tweeted. 2018 will be a great year for America! As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year. 2018 will be a great year for America! Donald J. Trump (@realDonaldTrump) December 31, 2017Trump s tweet went down about as welll as you d expect.What kind of president sends a New Year s greeting like this despicable, petty, infantile gibberish? Only Trump! His lack of decency won t even allow him to rise above the gutter long enough to wish the American citizens a happy new year! Bishop Talbert Swan (@TalbertSwan) December 31, 2017no one likes you Calvin (@calvinstowell) December 31, 2017Your impeachment would make 2018 a great year for America, but I ll also accept regaining control of Congress. Miranda Yaver (@mirandayaver) December 31, 2017Do you hear yourself talk? When you have to include that many people that hate you you have to wonder? Why do the they all hate me? Alan Sandoval (@AlanSandoval13) December 31, 2017Who uses the word Haters in a New Years wish?? Marlene (@marlene399) December 31, 2017You can t just say happy new year? Koren pollitt (@Korencarpenter) December 31, 2017Here s Trump s New Year s Eve tweet from 2016.Happy New Year to all, including to my many enemies and those who have fought me and lost so badly they just don t know what to do. Love! Donald J. Trump (@realDonaldTrump) December 31, 2016This is nothing new for Trump. He s been doing this for years.Trump has directed messages to his enemies and haters for New Year s, Easter, Thanksgiving, and the anniversary of 9/11. pic.twitter.com/4FPAe2KypA Daniel Dale (@ddale8) December 31, 2017Trump s holiday tweets are clearly not presidential.How long did he work at Hallmark before becoming President? Steven Goodine (@SGoodine) December 31, 2017He s always been like this . . . the only difference is that in the last few years, his filter has been breaking down. Roy Schulze (@thbthttt) December 31, 2017Who, apart from a teenager uses the term haters? Wendy (@WendyWhistles) December 31, 2017he s a fucking 5 year old Who Knows (@rainyday80) December 31, 2017So, to all the people who voted for this a hole thinking he would change once he got into power, you were wrong! 70-year-old men don t change and now he s a year older.Photo by Andrew Burton/Getty Images.	News	December 31, 2017

A tibble: 1 × 4

title <chr>	text <chr>	subject <chr>	date <chr>
As U.S.	WASHINGTON (Reuters) - The head of a conservative Republican faction in the U.S. Congress, who voted this month for a huge expansion of the national debt to pay for tax cuts, called himself a "fiscal conservative" on Sunday and urged budget restraint in 2018. In keeping with a sharp pivot under way among Republicans, U.S. Representative Mark Meadows, speaking on CBS' "Face the Nation," drew a hard line on federal spending, which lawmakers are bracing to do battle over in January. When they return from the holidays on Wednesday, lawmakers will begin trying to pass a federal budget in a fight likely to be linked to other issues, such as immigration policy, even as the November congressional election campaigns approach in which Republicans will seek to keep control of Congress. President Donald Trump and his Republicans want a big budget increase in military spending, while Democrats also want proportional increases for non-defense "discretionary" spending on programs that support education, scientific research, infrastructure, public health and environmental protection. "The (Trump) administration has already been willing to say: 'We're going to increase non-defense discretionary spending ... by about 7 percent,'" Meadows, chairman of the small but influential House Freedom Caucus, said on the program. "Now, Democrats are saying that's not enough, we need to give the government a pay raise of 10 to 11 percent. For a fiscal conservative, I don't see where the rationale is. ... Eventually you run out of other people's money," he said. Meadows was among Republicans who voted in late December for their party's debt-financed tax overhaul, which is expected to balloon the federal budget deficit and add about $1.5 trillion over 10 years to the $20 trillion national debt. "It's interesting to hear Mark talk about fiscal responsibility," Democratic U.S. Representative Joseph Crowley said on CBS. Crowley said the Republican tax bill would require the United States to borrow $1.5 trillion, to be paid off by		

Data Decription and Data Cleaning

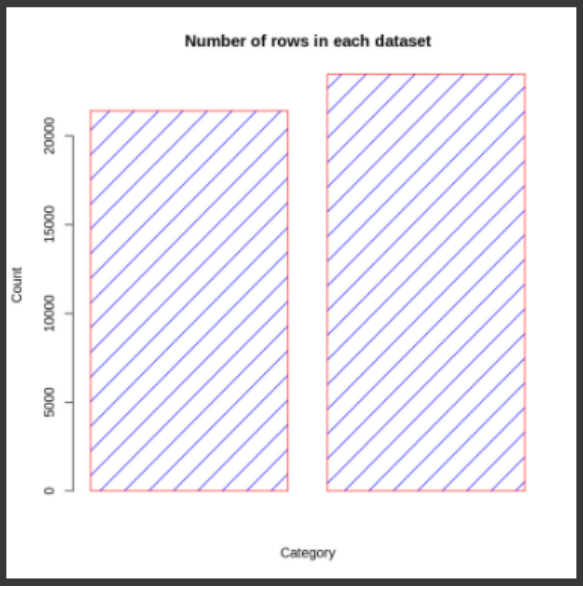
The data required for this project is present in two datafiles of csv format. The files are namely true.csv and fake.csv. The true.csv files contains news that are true and similarly for Fake.csv which contains fake news.

Number of columns

```
[ ]  print("The number of columns in true.csv")
     dim(true)
     print("The number of columns in fake.csv")
     dim(fake)
```

```
[1] "The number of columns in true.csv"
21417 · 4
[1] "The number of columns in fake.csv"
23481 · 4
```

```
barplot(c(nrow(true) , nrow(fake)) ,
        main="Number of rows in each dataset",
        xlab="Category",
        ylab="Count",
        border="red",
        col="blue",
       density = 5)
```



Both the datasets are balanced.

Columns datatypes

```
[ ] glimpse(true)
    glimpse(fake)
```

```
Rows: 21,417
Columns: 4
$ title   <chr> "As U.S. budget fight looms, Republicans flip their fiscal sc…
$ text    <chr> "WASHINGTON (Reuters) - The head of a conservative Republican…
$ subject <chr> "politicsNews", "politicsNews", "politicsNews", "politicsNews…
$ date    <chr> "December 31, 2017", "December 29, 2017", "December 31, 2017"…
Rows: 23,481
Columns: 4
$ title   <chr> "Donald Trump Sends Out Embarrassing New Year’s Eve Message; …
$ text    <chr> "Donald Trump just couldn t wish all Americans a Happy New Ye…
$ subject <chr> "News", "News", "News", "News", "News", "News", "News", "News…
$ date    <chr> "December 31, 2017", "December 31, 2017", "December 30, 2017"…
```

Are any NA values present?

```
[ ]  sum(is.na(true))
     sum(is.na(fake))
```

```
1
630
```

```
[ ]  ## Percentage of total dataset

     sum(is.na(true))/nrow(true)*100
     sum(is.na(fake))/nrow(fake)*100
```

```
0.0046691880282019
2.68302031429667
```

As we can see that NA values are present and when compared to the total numbers of rows they are only 0.004% and 2.68% of the total dataset. Instead of predicting them we can just drop them because large amount of data is not lost.

```
[ ]  true <- true %>% drop_na()
     fake <- fake %>% drop_na()
```

```
[ ]  dim(true)
     dim(fake)
```

```
21416 · 4
22851 · 4
```

Summary of datasets

```
[ ]  summary(fake)
     summary(true)
```

```
     title              text              subject            date
 Length:22851       Length:22851       Length:22851       Length:22851
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
    title              text              subject            date
 Length:21416       Length:21416       Length:21416       Length:21416
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

Merging datasets for further preprocessing

```
[ ]  fake$y <- 0
     true$y <- 1
     news <- bind_rows(fake, true)

     ## since the y column is of categorical type and the models will consider it of numerical if we do not convert it to factor.
     ## The same applies for the subject columns
     news$y <- as.factor(news$y)
     news$subject <- as.factor(news$subject)
```

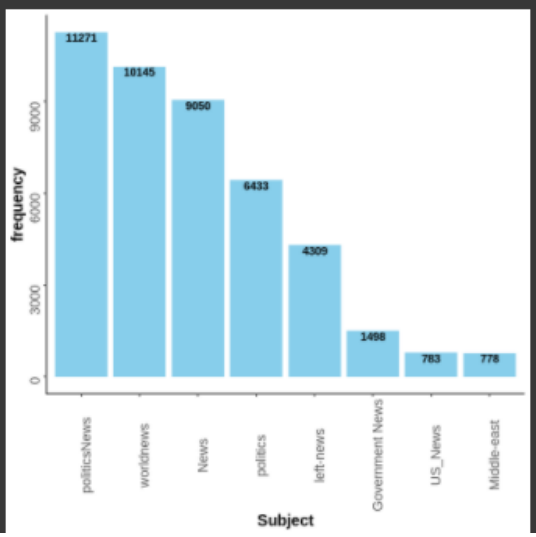
Preprocessing

```
[ ]  # News count by each Subject
     news %>% group_by(subject) %>% count() %>% arrange(desc(n))
```

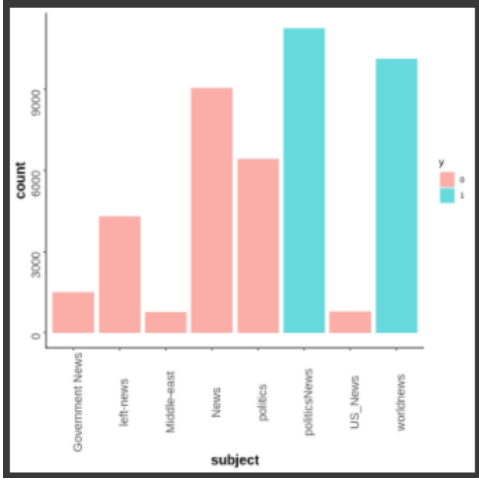
A grouped_df: 8 × 2

subject	n
<fct>	<int>
politicsNews	11271
worldnews	10145
News	9050
politics	6433
left-news	4309
Government News	1498
US_News	783
Middle-east	778

```
news %>%
  group_by(subject) %>%
  count(sort = TRUE) %>%
  rename(freq = n) %>%
  ggplot(aes(x = reorder(subject, -freq), y = freq)) +
  geom_bar(stat = 'identity', fill = 'skyblue') +
  theme_classic() +
  xlab('Subject') +
  ylab('frequency') +
  geom_text(aes(label = freq), vjust = 1.2, fontface = 'bold') +
  theme(axis.title = element_text(face = 'bold', size = 15),
        axis.text = element_text(size = 13, angle = 90))
```



```
## Categoty wise news subject plot

ggplot(news, aes(x = subject , fill = y)) +
  geom_bar(position = 'dodge', alpha = 0.6) +
  theme_classic() +
  theme(axis.title = element_text(face = 'bold', size = 15),
        axis.text = element_text(size = 13, angle = 90))
```



A news article is the combination of its heading and the text written below the heading. So we merge the title and text column into one. Reducing the dimension of the dataset.

We can also see that the subject column is biased in nature and will lead to decreased accuracy. So we drop that column

```
[ ]  y = news$y
     news$text <- paste(news$title , news$text , sep = ' ')
     news <- cbind(news["text"] , y)
     glimpse(news)
```

```
     Rows: 44,267
     Columns: 2
     $ text <chr> "Donald Trump Sends Out Embarrassing New Year's Eve Message; Thi…
     $ y    <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
```

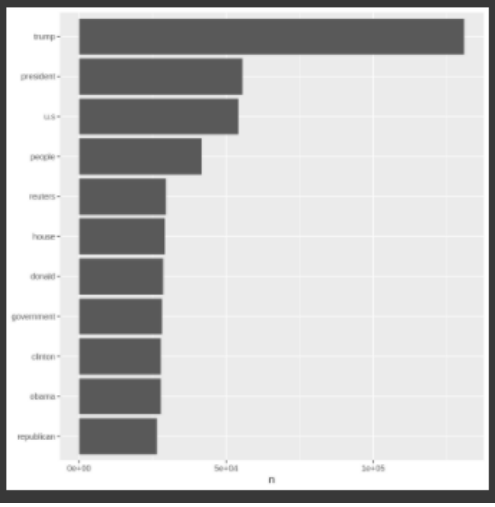
N-Gram Analysis

Uni Gram Analysis

```
[ ] tokeniztion_df <- news %>% unnest_tokens(word, text)
    tokeniztion_df <- tokeniztion_df %>% anti_join(stop_words)
```

```
Joining, by = "word"
```

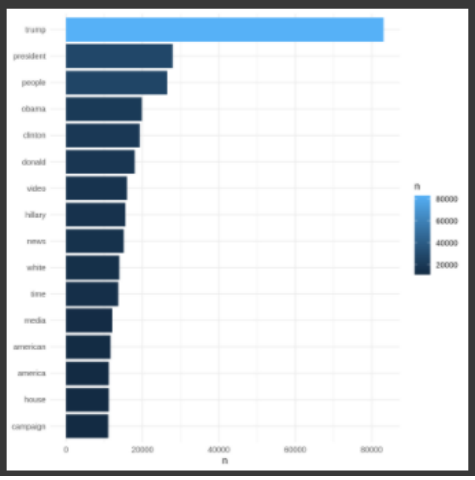
```
tokeniztion_df %>% count(word, sort = TRUE) %>% filter(n > 25000) %>% mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL)
```




```
[ ] fake_news <- news%>%filter(y == 0)
    true_news <- news%>%filter(y == 1)
```

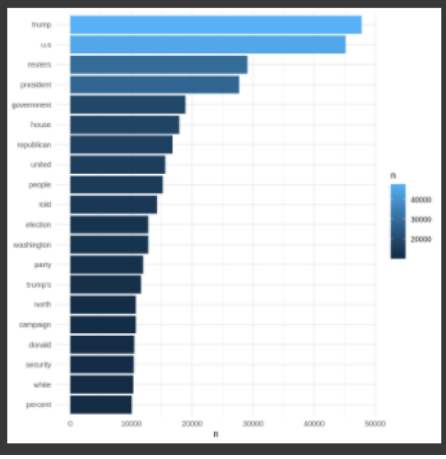
```
#Unigram - Fake_News
tokeniztion_df_fake <- fake_news %>% unnest_tokens(word, text)
tokeniztion_df_fake <- tokeniztion_df_fake %>% anti_join(stop_words)
tokeniztion_df_fake %>% count(word, sort = TRUE) %>% filter(n > 10000) %>% mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word , fill = n)) +
  geom_col() +
  labs(y = NULL) +
  theme_minimal()
```

```
Joining, by = "word"
```



```
#Unigram - True_News
tokeniztion_df_true <- true_news %>% unnest_tokens(word, text)
tokeniztion_df_true <- tokeniztion_df_true %>% anti_join(stop_words)
tokeniztion_df_true %>% count(word, sort = TRUE) %>% filter(n > 10000) %>% mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word,fill = n)) +
  geom_col() +
  labs(y = NULL)+
  theme_minimal()
```

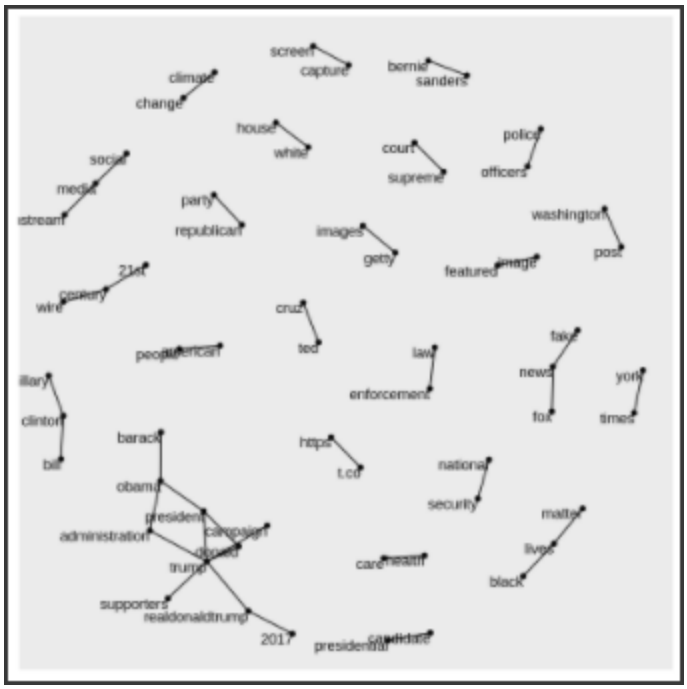
```
Joining, by = "word"
```



Bi Gram Analysis

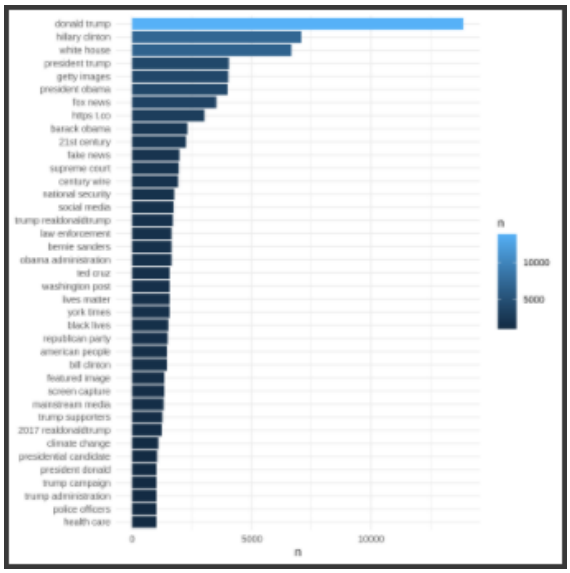
Category Wise

```
df_bigrams <-  fake_news %>% unnest_tokens(bigram, text, token = "ngrams", n = 2)
bigrams_separated <- df_bigrams %>% separate(bigram, c("word1", "word2"), sep = " ") %>% filter(!word1 %in% stop_words$word) %>% filter(!word2 %in% stop_words$word)
bigram_counts <- bigrams_separated %>%  count(word1, word2, sort = TRUE)
bigram_graph <- bigram_counts %>% filter(n > 1000) %>% graph_from_data_frame()
set.seed(2017)
ggraph(bigram_graph, layout = "fr") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```

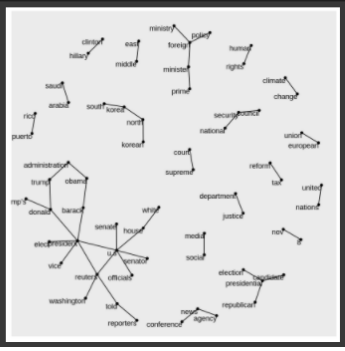


```
[ ]  fak_bigram <- data.frame(word <- paste(bigram_counts$word1 , bigram_counts$word2 , sep = " "))
     fak_bigram$n <- bigram_counts$n
     fak_bigram <- fak_bigram %>% filter(n > 1000)
     names(fak_bigram) <- c("word" , "n")
```

```
[ ]  fak_bigram %>% arrange(desc(n)) %>% mutate(word = reorder(word, n)) %>%
       ggplot(aes(n, word,fill = n)) +
       geom_col() +
       labs(y = NULL) +
       theme_minimal()
```

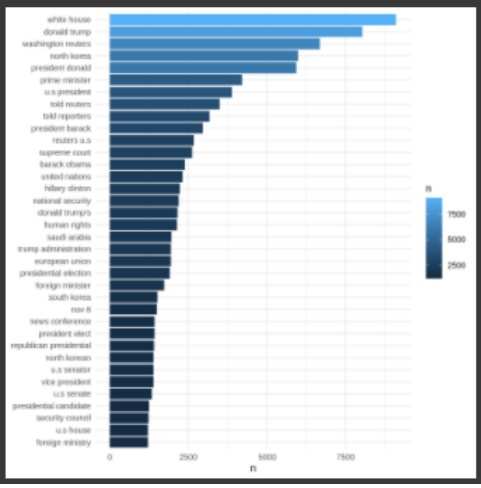



```
[ ]  tdf_bigrams <-  true_news %>% unnest_tokens(bigram, text, token = "ngrams", n = 2)
     tbigrams_separated <- tdf_bigrams %>% separate(bigram, c("word1", "word2"), sep = " ") %>% filter(!word1 %in% stop_words$word) %>% filter(!word2 %in% stop_words$word)
     tbigram_counts <- tbigrams_separated %>%  count(word1, word2, sort = TRUE)
     tbigram_graph <- tbigram_counts %>% filter(n > 1000) %>% graph_from_data_frame()
     set.seed(2017)
     ggraph(tbigram_graph, layout = "fr") +
       geom_edge_link() +
       geom_node_point() +
       geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```



```
[ ]  tru_bigram <- data.frame(paste(tbigram_counts$word1 , tbigram_counts$word2 , sep = " "))
     tru_bigram$n <- tbigram_counts$n
     ttru_bigram <- tru_bigram %>% filter(n > 1200)
     names(ttru_bigram) <- c("word" , "n")
```

```
[ ]  ttru_bigram %>% arrange(desc(n)) %>% mutate(word = reorder(word, n)) %>%
       ggplot(aes(n, word , fill = n)) +
       geom_col() +
       labs(y = NULL) +
       theme_minimal()
```



We implement various test preprocessing technique

- to lowercase
- remove numbers
- remove punctutaions
- remove stopwords
- remove whitespaces
- Lemmatization
- Tokenization

```
[ ]  ## Importing data
     fake <- read_csv('Fake.csv')
     true <- read_csv('True.csv')
     ##Dropping NA rows
     true_news <- true_news %>% drop_na()
     fake_news <- fake_news %>% drop_na()
     ## Merging Datasets
     fake_news$type <- 0
     true_news$type <- 1
     news <- bind_rows(fake_news, true_news)
     news$type <- as.factor(news$type)
     type = news$type
     news$text <- paste(news$title , news$text , sep = ' ')
     news <- cbind(news["text"] , type)
     data <- news
```

```
── Column specification ────────────────────────────────────────────────────────
cols(
  title = col_character(),
  text = col_character(),
  subject = col_character(),
  date = col_character()
)


── Column specification ────────────────────────────────────────────────────────
cols(
  title = col_character(),
  text = col_character(),
  subject = col_character(),
  date = col_character()
)
```

```
[ ] data <- news[sample(nrow(data)),]
```

```
[ ] ##Preprocessing
    doc <- VCorpus(VectorSource(data$text))
    doc <- tm_map(doc, removePunctuation)
    doc <- tm_map(doc, removeNumbers)
    doc <- tm_map(doc, content_transformer(tolower))
    doc <- tm_map(doc, removeWords, stopwords("english"))
    doc <- tm_map(doc, stripWhitespace)
    doc <- tm_map(doc, content_transformer(lemmatize_strings))
```

```
## Data Preparation
dtm <- DocumentTermMatrix(doc)
dtm_clean <- removeSparseTerms(dtm, sparse = 0.99)
dtm_mat <- as.matrix(dtm_clean)
y_prediction = data$type
dtm_mat <- cbind(dtm_mat,y_prediction)
dtm_df <- as.data.frame(dtm_mat)
```

```
[ ] summary(dtm_df$y_prediction)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.484   2.000   2.000
```

```
[ ] dtm_df$y_prediction <- ifelse(dtm_df$y_prediction == 2, 1, 0)
    dtm_df$y_prediction <- as.factor(dtm_df$y_prediction)
```

```
[ ] ## Train Test Split
    set.seed(2020)
    index <- sample(nrow(dtm_df), nrow(dtm_df)*0.8, replace = FALSE)
    train_set <- dtm_df[index,]
    test_set <- dtm_df[-index,]
    names(train_set) <- make.names(names(train_set))
    names(test_set) <- make.names(names(test_set))
```

```r
## Fitting Model
#Random Forest
k <- round(sqrt(ncol(train_set)-1))
clf_rf <- randomForest(formula = y_prediction ~ .,data = train_set,ntree = 5 ,mtry = k,method = 'class')

#Naive Bayes
clf_nb <- naive_bayes(y_prediction ~ ., data = train_set)
```

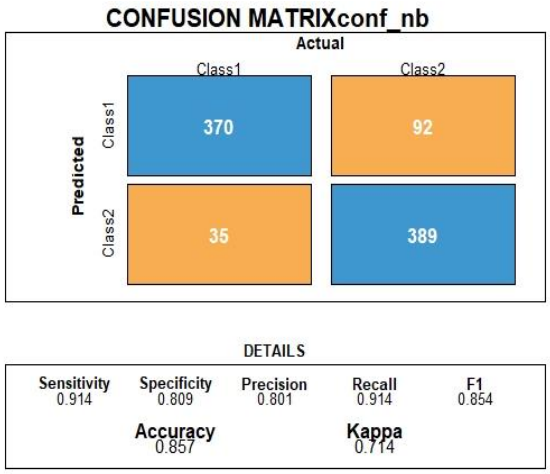
```r
##Meta Classifier Stacking
# Predicted values
train_set$pred_nb <- as.factor(predict(clf_nb, type = 'class'))
train_set$pred_rf <- as.factor(predict(clf_rf, type = 'response'))

# Predicted Values for test set
test_set$pred_nb <- as.factor(predict(clf_nb, newdata = test_set))
test_set$pred_rf <- as.factor(predict(clf_rf, newdata = test_set, type = 'response'))

#Stacking
train_set <- train_set[c("pred_nb" , "pred_rf" , "y_prediction")]
test_set <-  test_set[c("pred_nb" , "pred_rf" , "y_prediction")]
```

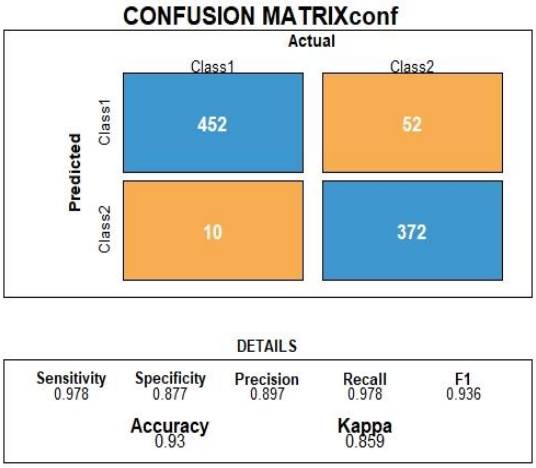
```r
##Logistics Regression
clf_lr <- glm(formula = y_prediction~.,  data = train_set, family=binomial(link="logit"))
```

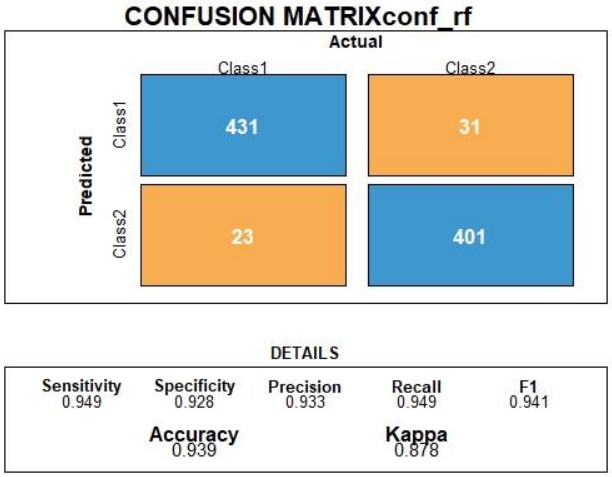
```r
test_set$pred_lr <- predict(clf_lr, newdata = test_set, type = 'response')
test_set$pred_lr <- ifelse(test_set$pred_lr > 0.5,1,0)
test_set$pred_lr <- as.factor(test_set$pred_lr)
```

```r
# Confussion Matrix
conf <- confusionMatrix(reference = test_set$y_prediction, data = test_set$pred_lr)
conf_nb <- caret::confusionMatrix(test_set$y_prediction, test_set$pred_nb)
conf_rf <- caret::confusionMatrix(test_set$y_prediction, test_set$pred_rf)
```

```r
draw_confusion_matrix(conf)
draw_confusion_matrix(conf_nb)
draw_confusion_matrix(conf_rf)
```






CONFUSION MATRIXconf_rf
Actual
Class1
Class2
Predicted
Class1
Class2
431
31
23
401

DETAILS
Sensitivity
0.949
Specificity
0.928
Precision
0.933
Recall
0.949
F1
0.941
Accuracy
0.939
Kappa
0.878