PROJECT REPORT

(Semester 19202 Jan 2020-May 2020)

Fake News Detection

Under the Guidance of

Dr. Divya Bansal
Professor, CSE
Punjab Engineering College, Chandigarh

Submitted By

Lovedeep Singh, SID-16103104
Kanishk Gautam, SID-16103118
Mehul Narang, SID- 16103057
Akshat Kasana, SID- 16103081
Abhinav Thakur, SID- 16103093

Department of Computer Science and Engineering
Punjab Engineering College, Chandigarh
(Deemed University)

Jan 2020 to May 2020

PROJECT REPORT

(Semester 19202 Jan 2020-May 2020)

Fake News Detection

Under the Guidance of

Dr. Divya Bansal
Professor, CSE
Punjab Engineering College, Chandigarh

Submitted By

Lovedeep Singh, SID-16103104
Kanishk Gautam, SID-16103118
Mehul Narang, SID- 16103057
Akshat Kasana, SID- 16103081
Abhinav Thakur, SID- 16103093

Department of Computer Science and Engineering
Punjab Engineering College, Chandigarh
(Deemed University)

Jan 2020 to May 2020

DECLARATION

We hereby declare that the project work entitled 'Fake News Detection' is an authentic record of our own work carried as requirements of the major project for the award of degree of B.Tech. Computer Science and Engineering, Punjab Engineering College, Chandigarh, under the guidance Dr. Divya Singla, Professor, CSE during Jan 2020 to May 2020.

Date: ________________

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

Dr. Divya Bansal
Professor
Computer Science and Engineering
Punjab Engineering College, Chandigarh

ACKNOWLEDGEMENT

Training is an agglomeration of theoretical, practical and technical concepts that enhances our skills in the field of technology. Training under renowned and knowledgeable mentors can yield prolific results wherein the cachet and technical skills are imparted.

We would like to express our deep sense of gratitude towards Dr. Divya Bansal for providing us an opportunity to work in a challenging project. It is our radiant sentiment to place on our record our best regards, deepest sense of gratitude to Dr. Divya Bansal, Professor for their careful and precious guidance which were extremely valuable for our study both theoretically and practically.

We are also thankful to Ms. Shubhangi for guiding us in the phase of the project that helped us to complete the module well within time thereby giving us the opportunity to work well on other modules during our Major Project.

We thank profusely all the colleagues for their kind help, friendly nature, timely suggestions and co-operation throughout our Major Project.

(Lovedeep Singh, SID-16103104
Kanishk Gautam, SID-16103118
Mehul Narang, SID-16103057
Akshat Kasana, SID-16103081
Abhinav Thakur, SID-16103093)

ABSTRACT

In recent times fake news for various commercial and political purposes has been appearing in large numbers and widespread in the online world. With deceptive words, online social network users can get infected by this fake news easily, which has brought about tremendous effects on the offline society already. An important goal in improving the trustworthiness of information in online social networks is to identify the fake news timely. This project aims at investigating the principles, methodologies and algorithms for detecting fake news articles, creators and subjects from online news articles, social networks and evaluating the corresponding performance. This project tries to addresses the challenges introduced by the unknown characteristics of fake news and diverse connections among news articles, creators and subjects. Based on a set of explicit and latent features extracted from the textual information, we initially explore elementary machine learning models to see the performance of these on news articles, creators and subjects simultaneously. We also explore areas in Deep learning intersecting with NLP that could help us to tackle Fake News.

Chapter-1 INTRODUCTION

Figure 1.1 Introductory Illustration

Fake news is a form of news consisting of deliberate disinformation or hoaxes spread via traditional news media (print and broadcast) or online social media. Digital news has brought back and increased the usage of fake news. The news is then often reverberated as misinformation in social media but occasionally finds its way to the mainstream media as well.

Fake news is written and published usually with the intent to mislead in order to damage an agency, entity, or person, and/or gain financially or politically, often using sensationalist, dishonest, or outright fabricated headlines to increase readership. Similarly, clickbait stories and headlines earn advertising revenue from this activity.

The relevance of fake news has increased in post-truth politics. For media outlets, the ability to attract viewers to their websites is necessary to generate online advertising revenue. Publishing a story with false content that attracts users benefits advertisers and improves ratings. Easy access to online advertisement revenue, increased political polarization and the popularity of social media, primarily the Facebook News Feed, have all been implicated in the spread of fake news, which competes with legitimate news stories. Hostile government actors have also been implicated in generating and propagating fake news, particularly during elections.

Fake news undermines serious media coverage and makes it more difficult for journalists to cover significant news stories. An analysis by BuzzFeed found that the top 20 fake news stories about the 2016 U.S. presidential election received more engagement on Facebook than the top 20 election stories from 19 major media outlets. Anonymously-hosted fake news websites lacking known publishers have also been criticized, because they make it difficult to prosecute sources of fake news for libel.

Fake news is a neologism often used to refer to fabricated news. This type of news, found in traditional news, social media or fake news websites, has no basis in fact, but is presented as being factually accurate.

The intent and purpose of fake news is important. In some cases, what appears to be fake news may be news satire, which uses exaggeration and introduces non-factual elements that are intended to amuse or make a point, rather than to deceive. Propaganda can also be fake news. Some researchers have highlighted that "fake news" may be distinguished not just by the falsity of its content, but also the "character of [its] online circulation and reception".

Claire Wardle of *First Draft News* identifies seven types of fake news:

1. satire or parody ("no intention to cause harm but has potential to fool")
2. false connection ("when headlines, visuals or captions don't support the content")
3. misleading content ("misleading use of information to frame an issue or an individual")
4. false context ("when genuine content is shared with false contextual information")
5. impostor content ("when genuine sources are impersonated" with false, made-up sources)
6. manipulated content ("when genuine information or imagery is manipulated to deceive", as with a "doctored" photo)
7. fabricated content ("new content is 100% false, designed to deceive and do harm")

Here are a few examples of fake news:

- Clickbait

- Propaganda
- Satire/parody
- Sloppy journalism
- Misleading headings
- Biased or slanted news

There are features of fake news and may help to identify and avoid instances of fake news.

The International Federation of Library Associations and Institutions (IFLA) published a summary in diagram form *(pictured at right)* to assist people in recognizing fake news. Its main points are:

1. Consider the source (to understand its mission and purpose)
2. Read beyond the headline (to understand the whole story)
3. Check the authors (to see if they are real and credible)
4. Assess the supporting sources (to ensure they support the claims)
5. Check the date of publication (to see if the story is relevant and up to date)
6. Ask if it is a joke (to determine if it is meant to be satire)
7. Review your own biases (to see if they are affecting your judgment)
8. Ask experts (to get confirmation from independent people with knowledge).

The International Fact-Checking Network (IFCN), launched in 2015, supports international collaborative efforts in fact-checking, provides training, and has published a code of principles. In 2017 it introduced an application and vetting process for journalistic organisations. One of IFCN's verified signatories, the independent, not-for-profit media journal *The Conversation*, created a short animation explaining its fact checking process, which involves "extra checks and balances, including blind peer review by a second academic expert, additional scrutiny and editorial oversight".

Beginning in the 2017 school year, children in Taiwan study a new curriculum designed to teach critical reading of propaganda and the evaluation of sources. Called "media literacy", the course provides training in journalism in the new information society.

Detecting fake news online

Fake news has become increasingly prevalent over the last few years, with over 100 incorrect articles and rumors spread incessantly just with regard to the 2016 United States presidential election. These fake news articles tend to come from satirical news websites or individual websites with an incentive to propagate false information, either as clickbait or to serve a purpose. Since they typically hope to intentionally promote incorrect information, such articles are quite difficult to detect. When identifying a source of information, one must look at many attributes, including but not limited to the content of the email and social media engagements. specifically, the language is typically more inflammatory in fake news than real articles, in part because the purpose is to confuse and generate clicks. Furthermore, modeling techniques such as n-gram encodings and bag of words have served as other linguistic techniques to determine the legitimacy of a news source. On top of that, researchers have determined that visual-based cues also play a factor in categorizing an article, specifically some features can be designed to assess if a picture was legitimate and provides more clarity on the news. There is also many social context features that can play a role, as well as the model of spreading the news. Websites such as “altnews.in” try to detect this information manually, while certain universities are trying to build mathematical models to do this themselves.

In this chapter we discussed about the basic sense of fake news, its forms, how it progressed over time and how we can detect fake news. We also touched upon the modern approach of modeling techniques to determine the legitimacy of news.

In the next chapter, we discuss the motivation behind this project. We also elaborate upon the Literature Survey and our Area of focus.

Chapter-2 BACKGROUND

Fake News Detection has been a hot area for a long time and has garnered more focus in recent times. With the increased power of computation, better tools for manage huge corpus of data using Big Data Techniques and inventions of deep learning models over elementary models of machine learning, the possibilities in his area have increased manifold. We discuss the motivation behind this project elaborating the impact of fake news on our society. We then present the research papers we swiveled through to get thorough about the prevailing techniques in this field. Then we elaborate our areas of focus and discuss upon them.

2.1 Motivation

Due to extensive spread of fake news on social and news media, it has become an emerging research topic now a days that gained attention. In the news media and social media, the information is spread highspeed but without accuracy and hence detection mechanism should be able to predict news fast enough to tackle the dissemination of fake news. It has the potential for negative impacts on individuals and society. Therefore, detecting fake news on social media is important and also a technically challenging problem these days. We knew that Machine learning is helpful for building Artificial intelligence systems based on tacit knowledge because it can help us to solve complex problems due to real word data. The election results of US are a big example of the kind of impact of fake news has on the society. In India also, there is lot fake news propagated by different organizations which dupe people to think and act in a certain way. The organizations have varied motives behind spreading fake news, one could be paid revenue by third party for their ulterior moves, other could be to spread hate amongst people against the ruling government or create environment of tension between two bordering countries. In India only, there has been a lot of fake news recently on Kashmir Article 370, Balakot Air Strike, Ayodhya Verdict owing to the same reasons.

2.2 Literature Survey

We went through some good research papers to get the essence of on-going approaches in the field of fake news detection. In exact numbers, we swivelled through 5 research papers. These are listed in references as a separate section. In essence, these elaborate about the existing approaches in this area and also throw light on the characteristics of fake news. These further discuss about the shortcomings of the present approaches and suggest a future roadmap to build better models and make improvements in the existing techniques. We have tried to chunk about some targets from these and incorporated in the plan of our work. Specifically, we will try to explore methods to make the machine learning model dynamic so that it remains at par with the changing traits of fake news

2.3 Areas of Focus

We collected data topics relevant to India. We target recent topics where a lot of fake news was propagated like Article 370, Ayodhya Verdict, Balakot Air Strike and more. We also target areas relevant to India Politics like Indian Elections, GST, Demonetization and related spheres. The fake news is deliberately made such that it is very difficult for a person to make out a difference between fake and real news.

We tried and tested various combinations of features on different models ranging from classical ML classification models such as SVM to complex Deep Learning networks such as ANN

In this chapter, we have discussed about:

- **Motivation**
- **Literature Survey**
- **Areas of focus**

In the next chapter we are going to discuss about the proposed work of the project with minimal details.

Chapter 3- PROPOSED WORK

We begin with data collection for training our machine learning models. Target is to build a sufficiently labeled dataset for optimum training of ML models. Apart from swiveling through any relevant available datasets we explore News Websites and Social Websites for data scrapping. After data collection part, we clean data for any non-relevant traits and preprocess it for training of ML models. Then, we train different elementary models with this data and try to cite out the differences between these with possible reasoning. Finally, we explore and experiment with deep learning and ANNs for valuable insights and possible improvements.

3.1 Data

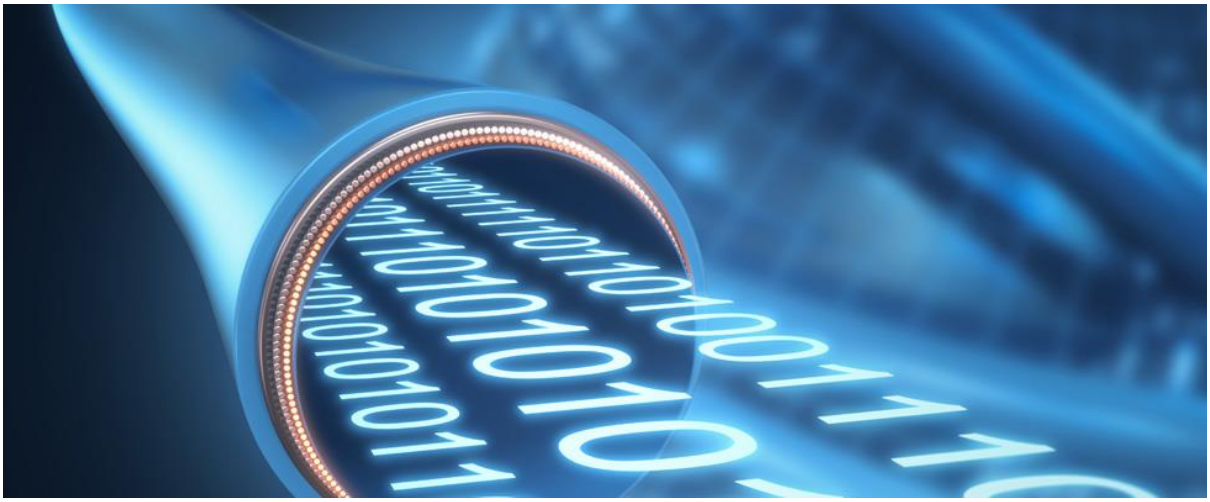


Figure 3.1 Data Illustration

Machine Learning (ML) algorithms learn from data. It is critical that you feed them the right data for the problem you want to solve. Even if you have good data, you need to make sure that it is in a useful scale, format and even that meaningful features are included. For data, we explore existing datasets for any relevant dataset that we can utilize without any major and only minimalistic modifications. After that we dive into the code for scrappers and explore various available APIs for extracting meaningful data from Social Websites and News Websites.

3.2 Machine Learning

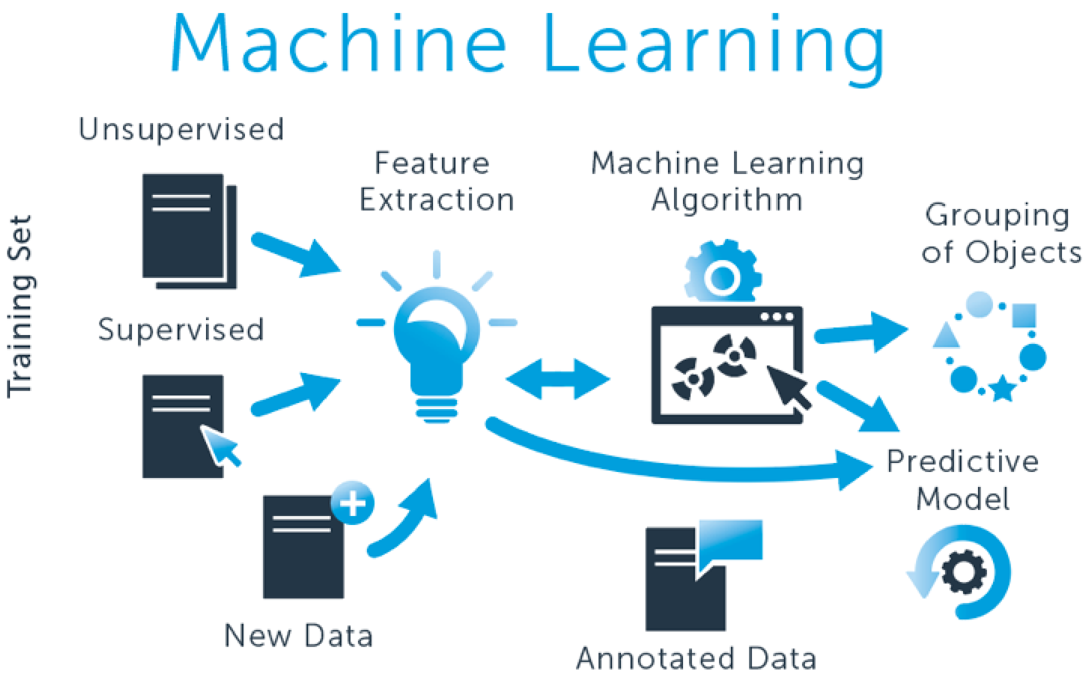


Figure 3.2 Basic ML Illustration

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. As main crux of fake news is ex, we must explore NLP and is fundamentals. After that, we explore trivial ML models for text classification. Then, we will also try incorporating features such as likes, comments and similar details prevalent in the social websites. Finally we experiment with Deep Learning and GANs

3.2.1 Text Classification

These mainly focus on extracting various features of text and after that incorporating of those features into classification models e.g. Decision tree, SVM, logistic regression, K nearest neighbor. At the end selection of best algorithm that performs well is a real time data driven rumor identification approach.

3.2.2 Trivial Machine Learning Models

Classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance. Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or features. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category. Terminology across fields is quite varied. In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables (or independent variables, regressors, etc.), and the categories to be predicted are known as outcomes, which are considered to be possible values of the dependent variable. In machine learning, the observations are often known as instances, the explanatory variables are termed features (grouped into a feature vector), and the possible categories to be predicted are classes. Other fields may use different terminology: e.g. in community ecology, the term "classification" normally refers to cluster analysis, i.e., a type of unsupervised learning, rather than the supervised learning described in this article. In all cases though, classifiers have a specific set of dynamic rules, which includes an interpretation procedure to handle vague or unknown values, all tailored to the type of inputs being examined. Since no single form of classification is appropriate for all data sets, a large toolkit of classification algorithms have been developed. The most commonly used include: Logistic regression, Naive Bayes classifier, Support vector machines, Least squares support vector machines, k-nearest neighbor, Decision trees and Random forests We will try hands on most of these models with the captured data and compare performances amongst these.

3.2.3 Deep Learning

Deep Learning

Deep learning imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support. One of the most common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data. To detect fake news, we could employ machine learning tools. The computational algorithm built into a computer model will process all records, find patterns in the data set and point out any anomaly detected by the pattern. Deep learning, a subset of machine learning, utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach. A traditional approach to detecting fake news might rely on the text that ensues, while a deep learning nonlinear technique would include time, geographic location, type of user and any other feature that is likely to point for fake news. The first layer of the neural network processes a raw data input like the text and passes it on to the next layer as output. The second layer processes the previous layer's information by including additional information like the source and passes on its result. The next layer takes the second layer's information and includes raw data like geographic location and makes the machine's pattern even better. This continues across all levels of the neuron network.

We try to explore deep learning for fake news detection. Using the detection system mentioned above with machine learning, one can create a deep learning example. If the machine learning system created a model with parameters built around the text, the deep-learning method can start building on the results offered by machine learning. Each layer of its neural network builds on its previous layer with added data like source, location, and a host of other features. Deep learning algorithms are trained to not just create patterns from all dataset, but also know when a pattern is signaling the need for a fake news investigation.

In this chapter we discussed in brief about the proposed work of project and the how the project shall progress.

The next chapter will give complete information about the implementation of the project, what models and techniques have been used

Chapter-4 IMPLEMENTATION DETAILS

4.1 Data

For data, we have explored available datasets from source like Kaggle. We have also extracted data from news websites and social websites using scrappers and available APIs. Each of these is discussed in detail below.

4.1.1 Available Datasets

Most of the available datasets for Fake News classification our based-on US affairs with a major chunk recently focused on elections won by Trump in 2016. We were not able to find exact dataset focused on our area of focus since most of the events are very recent. Although we explored other datasets for their size, attributes and results on various models for reference purposes.

4.1.2 Data Collection

Figure 4.1 Web Scrapping

We made programs for data scrapping from News Websites and Social Websites. We also made explored available APIs provided by Social Websites like tweepy for twitter and by News Websites like NYTimes.

Topics : Kashmir Article 370, Ayodhya Verdict, Indian Elections, GST, Demonetization, Maharashtra Elections.

4.1.2.1 News Websites

We have covered the following news sources:

NYTimes, Reuters, The Guardian, News API, Fauxy, BBC News and Times Of India. We have extracted data from these news sources for topics in our area of focus. All code has been written in Python and we have utilized Selenium and Beautiful Soup for web scrapping.

We here present the code written for Reuters

```
# coding: utf-8

# In[1]:


import os
import requests
from bs4 import BeautifulSoup
from selenium import webdriver
import numpy as np
import pandas as pd
import time



#print(html)


# In[80]:


chromedriver = "/Users/lovedeepsingh/Downloads/chromedriver"
os.environ["webdriver.chrome.driver"] = chromedriver
driver = webdriver.Chrome(chromedriver)
driver.get("https://in.reuters.com/search/")
elem = driver.find_element_by_id('newsSearchField')
elem.send_keys('kashmir article 370')
elem.send_keys(u'\ue007')

elm=driver.find_element_by_class_name('search-result-more-txt')

while True:
    elm = driver.find_element_by_class_name('search-result-more-txt')
    if 'search-result-more-txt search-result-no-more' in
elm.get_attribute('class'):
        break;
    elm.click()

time.sleep(2)
print(driver.current_url)
#url=driver.current_url+'/all'
url=driver.current_url

r1 = requests.get(url)
html = driver.page_source
#print(html)
coverpage = html
```

```
soup1 = BeautifulSoup(coverpage, 'html5lib')

coverpage_news = soup1.find_all('h3', class_='search-result-title')
number_of_articles=len(coverpage_news)
print(number_of_articles)

#int(coverpage_news)

#type(coverpage_news)

#number_of_articles = 5

news_contents = []
list_links = []
list_titles = []

for n in np.arange(0, number_of_articles):

    link = coverpage_news[n].find('a')['href']
    title = coverpage_news[n].find('a').get_text()
    list_titles.append(title)

    link = coverpage_news[n].find('a')['href']
    link="https://in.reuters.com/" + link
    list_links.append(link)

    article = requests.get(link)
    article_content = article.content

    soup_article = BeautifulSoup(article_content, 'html5lib')
    body = soup_article.find_all('div',
class_='StandardArticleBody_body')
    x = body[0].find_all('p')

    list_paragraphs = []
    for p in np.arange(0, len(x)):
        paragraph = x[p].get_text()
        list_paragraphs.append(paragraph)
        final_article = " ".join(list_paragraphs)

    news_contents.append(final_article)
    print(title)


# In[81]:

# df_features
df_features = pd.DataFrame(
     {'Article Content': news_contents
    })

# df_show_info
df_show_info = pd.DataFrame(
    {'Article Title': list_titles,
    'Article Content': news_contents})


# In[82]:
```

```
df_features

# In[83]:

df_show_info['label']=1

df_show_info.to_csv(r'kashmirarticle370.csv', index=False)
```

4.1.2.2 Social Websites

We made programs to scrap data from twitter and Instagram. For this we explored APIs like tweepy and twitter-scrapper, again the complete code has been written in python.

We here present the tweepy code for twitter:

```
import tweepy
import csv
import pandas as pd
####input your credentials here
consumer_key = '################'
consumer_secret = '#####################################'
access_token = '####################################'
access_token_secret = '##############################'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)
#####United Airlines
# Open/Create a file to append data
csvFile = open('a1213ede3w2qsw3.csv', 'a')
#Use csv Writer
csvWriter = csv.writer(csvFile)

csvWriter.writerow(["created at", "text", "user_screenname",
"user_verified" "user_location", "retweets_count", "retweet_content" ,
"favorites"])

for tweet in tweepy.Cursor(api.search,q="#article370",
                           lang="en",
                           since="2019-10-18").items():
    print (tweet.created_at, tweet)
    csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8'),
tweet.user.screen_name, tweet.user.verified, tweet.coordinates,
tweet.retweet_c
```

4.2 Classical Machine Learning

4.2.1 Pre-Processing NLP techniques

We have many techniques to transform natural language text into a machine-understandable form to train our ML models upon it. These include word embeddings, feature extractions, vector space transformations, and N-gram approaches. We use both Vector-Space models and feature extraction techniques. In vector-space, we focus on two techniques, TFIDF and Doc2Vec. In features, we focus on Linguistic features of the text such as Semantic Score, Punctuations, etc.

4.2.2 Word Embeddings

The goal is to produce a vector representation of each article. Before applying any transformation, we perform some basic pre-processing of the data. This includes removing stopwords, deleting special characters and punctuation, and converting all text to lowercase.

4.2.2.1 TFIDF

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP). TF-IDF was invented for document search and information retrieval. It works by increasing proportionally to the number of times a word appears in a document, but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times, since they don't mean much to that document in particular. However, if the word Bug appears many times in a document, while not appearing many times in others, it probably means that it's very relevant. For example, if what we're doing is trying to find out which topics some NPS responses belong to, the word Bug would probably end up being tied to the topic Reliability, since most responses containing that word would be about that topic.

TF-IDF for a word in a document is calculated by multiplying two different metrics:

- The **term frequency** of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by length of a document, or by the raw frequency of the most frequent word in a document.
- The **inverse document frequency** of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.
- So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1.

Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

To put it in more formal mathematical terms, the TF-IDF score for the word t in the document d from the document set D is calculated as follows:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

Where:

$$tf(t, d) = log(1 + freq(t, d))$$

$$idf(t, D) = log(\frac{N}{count(d \in D : t \in d)})$$

Machine learning with natural language is faced with one major hurdle – its algorithms usually deal with numbers, and natural language is, well, text. So we need to transform that text into numbers, otherwise known as text vectorization. It’s a fundamental step in the process of machine learning for analyzing text, and different vectorization algorithms will drastically affect end results, so you need to choose one that will deliver the results you’re hoping for. Once you’ve transformed words into numbers, in a way that’s machine learning algorithms can understand, the TF-IDF score can be fed to algorithms such as Naive Bayes and Support Vector Machines, greatly improving the results of more basic methods like word counts. Why does this work? Simply put, a word vector represents a document as a list of numbers, with one for each possible word of the corpus. Vectorizing a document is taking the text and creating one of these vectors, and the numbers of the vectors somehow represent the content of the text. TF-IDF enables us to gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm. It's useful to understand how TF-IDF works so that you can gain a better understanding of how machine learning algorithms function. While machine learning algorithms traditionally work better with numbers, TF-IDF algorithms help them decipher words by allocating them a numerical value or vector. This has been revolutionary for machine learning, especially in fields related to NLP such as text analysis. In text analysis with machine learning, TF-IDF algorithms help sort data into categories, as well as extract keywords. This means that simple, monotonous tasks, like tagging support tickets or rows of feedback and inputting data can be done in seconds.

4.2.2.2 Doc2Vec

Doc2Vec is a model developed in 2014 based on the existing Word2Vec model, which generates vector representations for words. Word2Vec represents documents by combining the vectors of the individual words, but in doing so it loses all word order information. Doc2Vec expands on Word2Vec by adding a “document vector” to the output representation, which contains some information about the document as a whole, and allows the model to learn some information about word order. Preservation of word order information makes Doc2Vec useful for our application, as we are aiming to detect subtle differences between text documents.

How Word2Vec works?

Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep nets can understand.

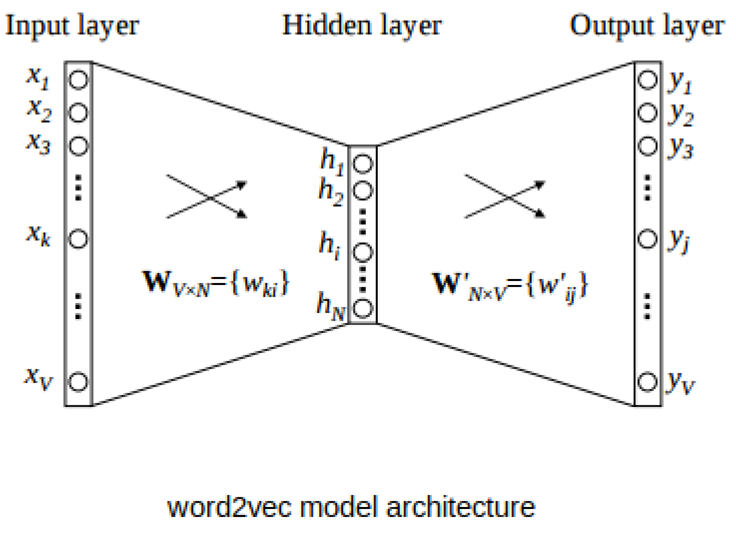


Figure 4.2.1 word2vec model architecture

Word2vec’s applications extend beyond parsing sentences in the wild. It can be applied just as well to genes, code, likes, playlists, social media graphs and other verbal or symbolic series in which patterns may be discerned.

Why? Because words are simply discrete states like the other data mentioned above, and we are simply looking for the transitional probabilities between those states: the likelihood that they will co-occur. So gene2vec, like2vec and follower2vec are all possible. With that in mind, the tutorial below will help you understand how to create neural embeddings for any group of discrete and co-occurring states.

The purpose and usefulness of Word2vec is to group the vectors of similar words together in vectorspace. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words. It does so without human intervention.

Given enough data, usage and contexts, Word2vec can make highly accurate guesses about a word's meaning based on past appearances. Those guesses can be used to establish a word's association with other words (e.g. "man" is to "boy" what "woman" is to "girl"), or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management.

The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words.

Measuring cosine similarity, no similarity is expressed as a 90 degree angle, while total similarity of 1 is a 0 degree angle, complete overlap

u.**v**/(|u||v|) - cosine similarity

Neural Word Embeddings

The vectors we use to represent words are called *neural word embeddings*, and representations are strange. One thing describes another, even though those two things are radically different. As Elvis Costello said: "Writing about music is like dancing about architecture." Word2vec "vectorizes" about words, and by doing so it makes natural language computer-readable – we can start to perform powerful mathematical operations on words to detect their similarities.

So a neural word embedding represents a word with numbers. It's a simple, yet unlikely, translation.

Word2vec is similar to an autoencoder, encoding each word in a vector, but rather than training against the input words through reconstruction, as a restricted Boltzmann machine does, word2vec trains words against other words that neighbor them in the input corpus.

It does so in one of two ways, either using context to predict a target word (a method known as continuous bag of words, or CBOW), or using a word to predict a target context, which is called skip-gram.

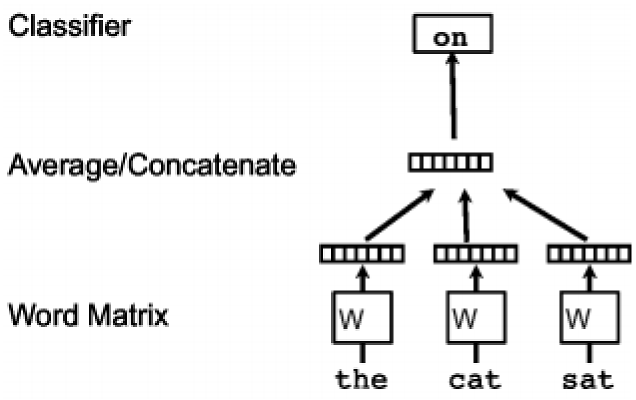


Figure 4.2.2 CBOW – Combined Bag of Words – Neural Word Embeddings

4.2.2 Implemented models and technique

We discuss the following ML models that we have used in our implementation phase one by one along with short theory and useful remarks. In all the models, we do different permutations and combinations using features and vector-based approaches. We have detailed the results in chapter 5. We briefly explain the models one by one below.

4.2.3 SVM

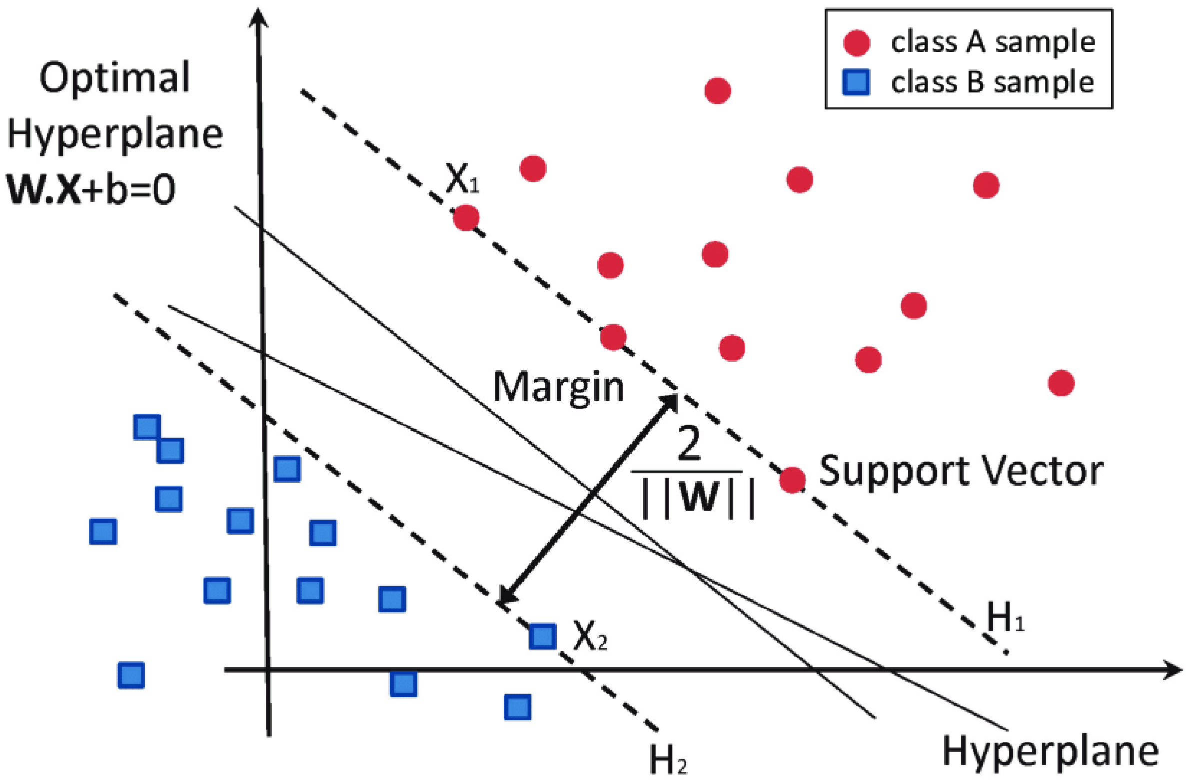


Figure 4.2.3 SVM

The original Support Vector Machine (SVM) was proposed by Vladimir N. Vapnik and Alexey Ya.Chervonenkis in 1963. But that model can only do linear classification so it doesn't suit for most of the practical problems. Later in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik introduced the kernel trick which enables the SVM for non-linear classification. That makes the SVM much powerful. We use the Radial Basis Function kernel in our project. The reason we use this kernel is that two Doc2Vec feature vectors will be close to each other if their corresponding documents are similar, so the distance computed by the kernel function should still represent the original distance. Since the Radial Basis Function is

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)$$

It correctly represents the relationship we desire and it is a common kernel for SVM. The main idea of the SVM is to separate different classes of data by the widest "street". This goal can be represented as the optimization problem

$$\arg\max_{w,b}\left\{\frac{1}{||w||}\min_{n}\left[t_n(w^T\phi(x_n)+b)\right]\right\}$$
$$s.t. \quad t_n(w^T\phi(x_n)+b)\geq 1, \qquad n=1,\dots N$$

Then we use the Lagrangian function to get rid of the constraints.

$$L(w,b,a)=\frac{1}{2}||w||^2-\sum_{n=1}^{N}a_n\left\{t_n(w^T\phi(x_n)+b)-1\right\}$$

where $a_n \geq 0, n=1,\dots N$.

Finally, we solve this optimization problem using the convex optimization tools provided by Python package CVXOPT.

4.2.4 Naive Bayes

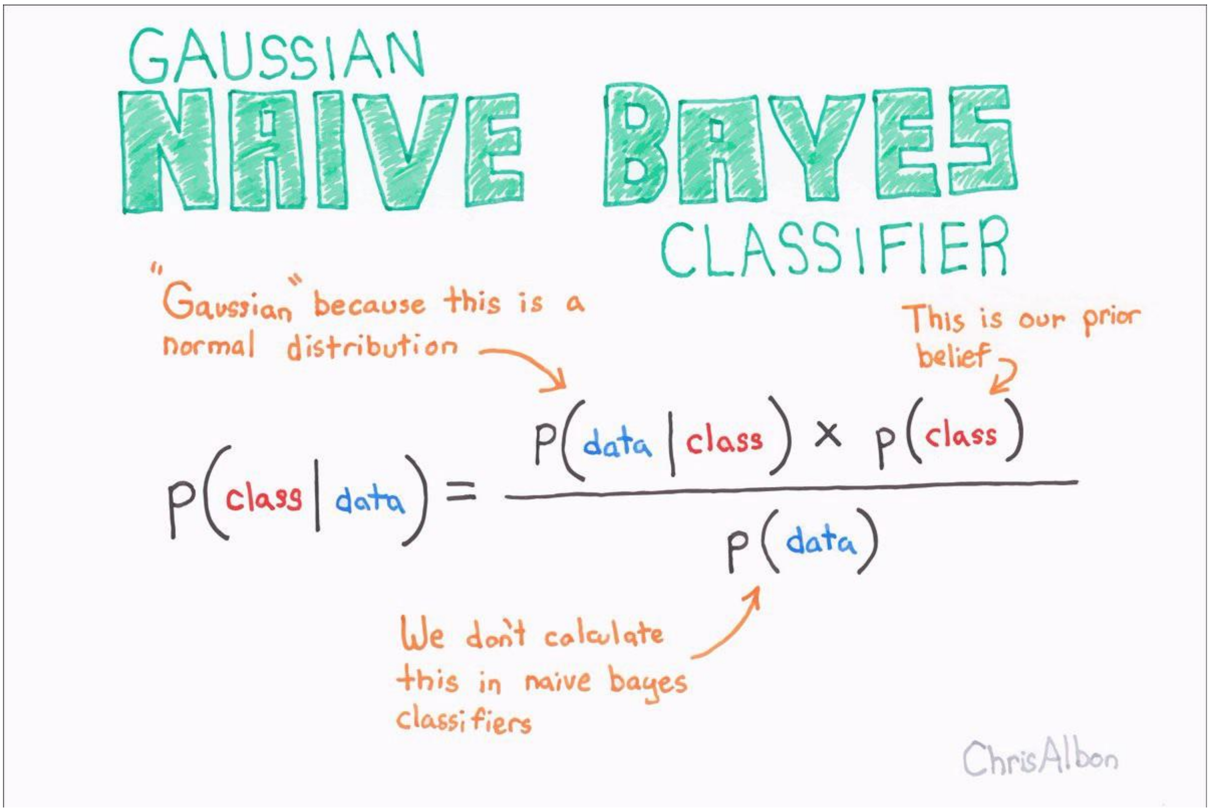


Figure 4.2.4 Naive Bayes

In order to get a baseline accuracy rate for our data, we implemented a Naive Bayes classifier. Specifically, we used the scikit-learn implementation of Gaussian Naive Bayes. This is one of the simplest approaches to classification, in which a probabilistic approach is used, with the assumption that all features are conditionally independent given the class label. As with the other model, we used the Doc2Vec embeddings described above. The Naive Bayes Rule is based on the Bayes' theorem

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Parameter estimation for naive Bayes models uses the method of maximum likelihood. The advantage here is that it requires only a small amount of training data to estimate the parameters.

4.2.5 KNN

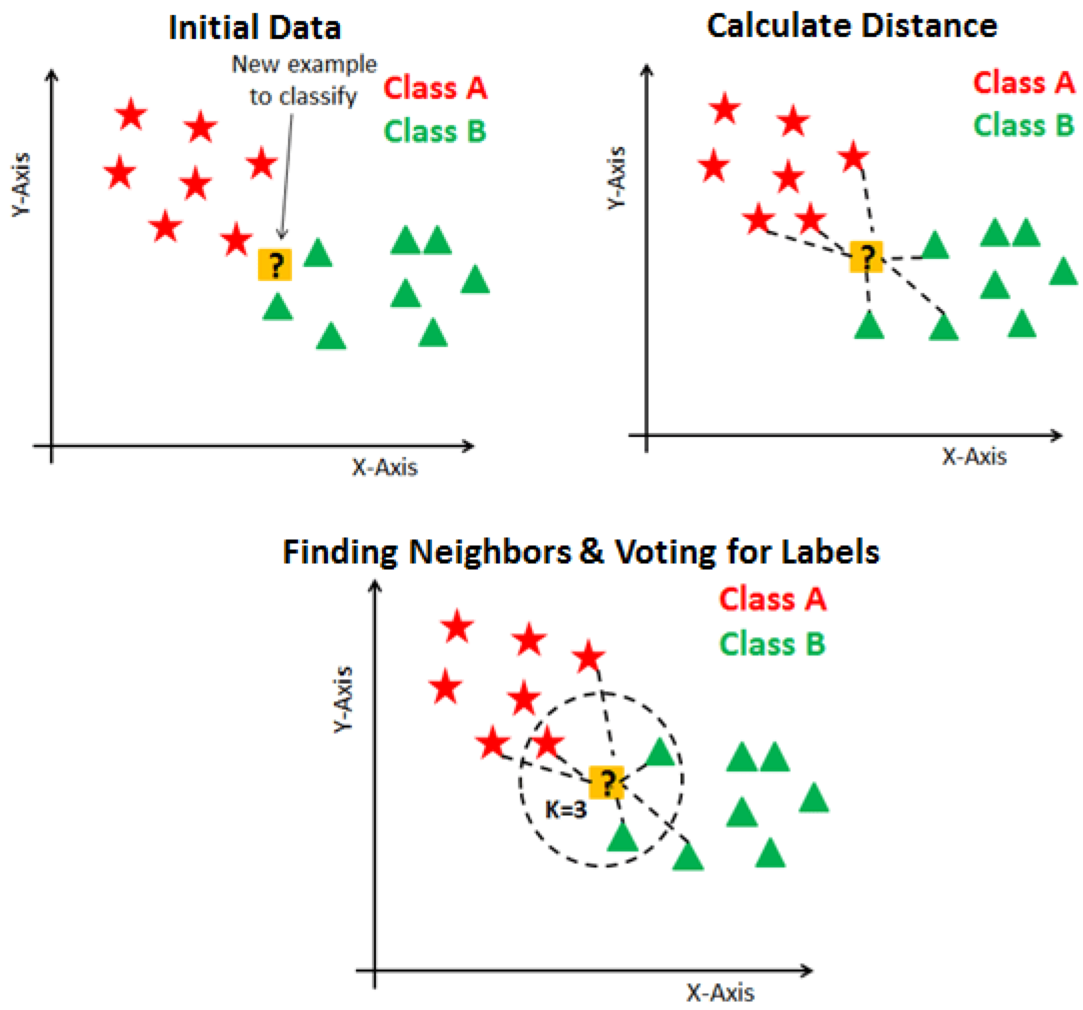


Figure 4.2.5 KNN

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance). In the context of gene expression microarray data, for example, k-NN has been employed with correlation coefficients, such as Pearson and Spearman, as a metric. Often, the classification accuracy of k-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis. A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the k nearest neighbors due to their large number. One way to overcome this problem is to weight the classification, taking into account the distance from the test point to each of its k nearest neighbors. The class (or value, in regression problems) of each of the k nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Another way to overcome skew is by abstraction in data representation. For example, in a self-organizing map (SOM), each node is a representative (a center) of a cluster of similar points, regardless of their density in the original training data. K-NN can then be applied to the SOM.

4.2.6 Logistic Regression

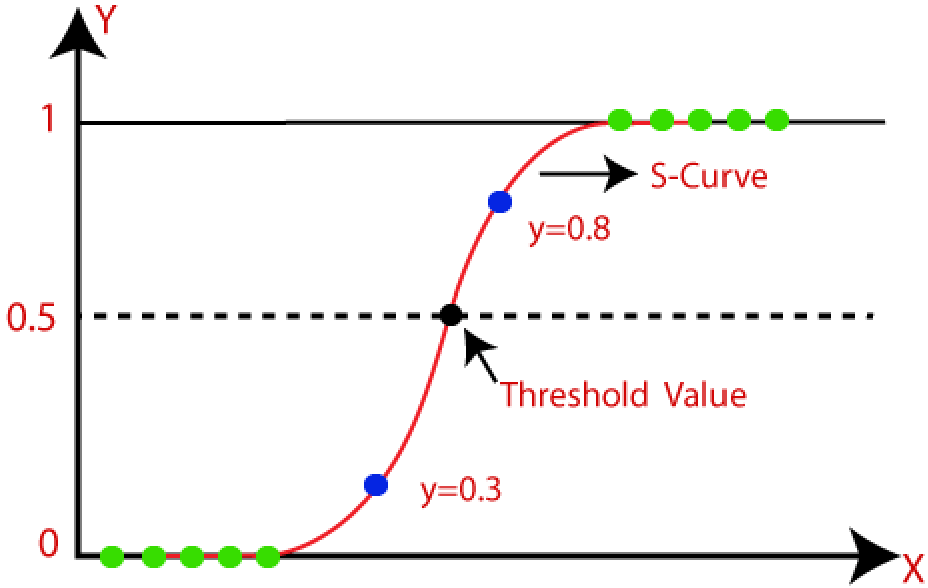


Figure 4.2.6 Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio. In a binary logistic regression model, the dependent variable has two levels (categorical). Outputs with more than two values are modeled by multinomial logistic regression and, if the multiple categories are ordered, by ordinal logistic regression (for example the proportional odds ordinal logistic model). The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier.

4.2.7 Decision Tree

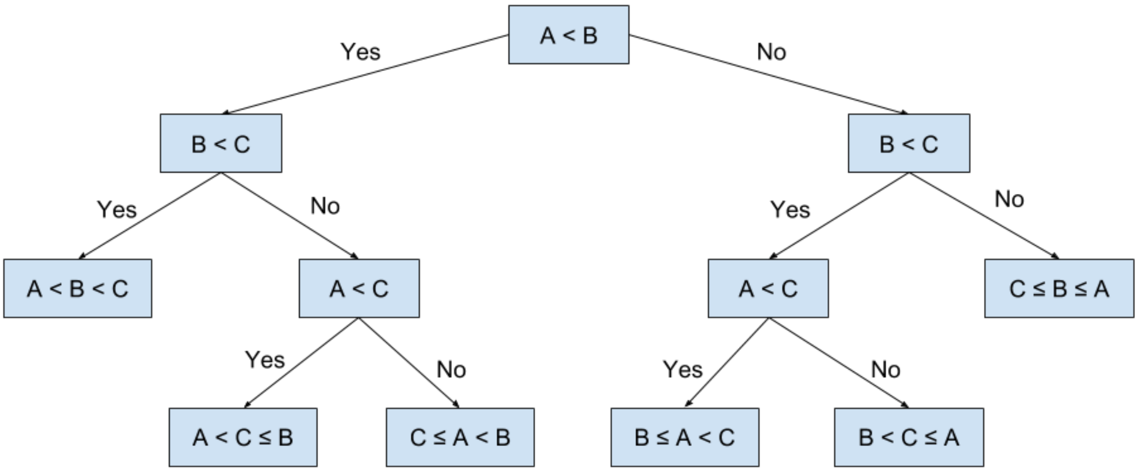


Figure 4.2.7 Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated. Decision trees are commonly used in operations research and operations management. If, in practice, decisions have to be taken online with no recall under incomplete knowledge, a decision tree should be paralleled by a probability model as a best choice model or online selection model algorithm. Another use of decision tree is as a descriptive means for calculating conditional probabilities. Decision trees, influence diagrams, utility functions, and other decision analysis tools and methods are taught to undergraduate students in schools of business, health economics, and public health, and are examples of operations research or management science methods.

4.2.8 Random Forest

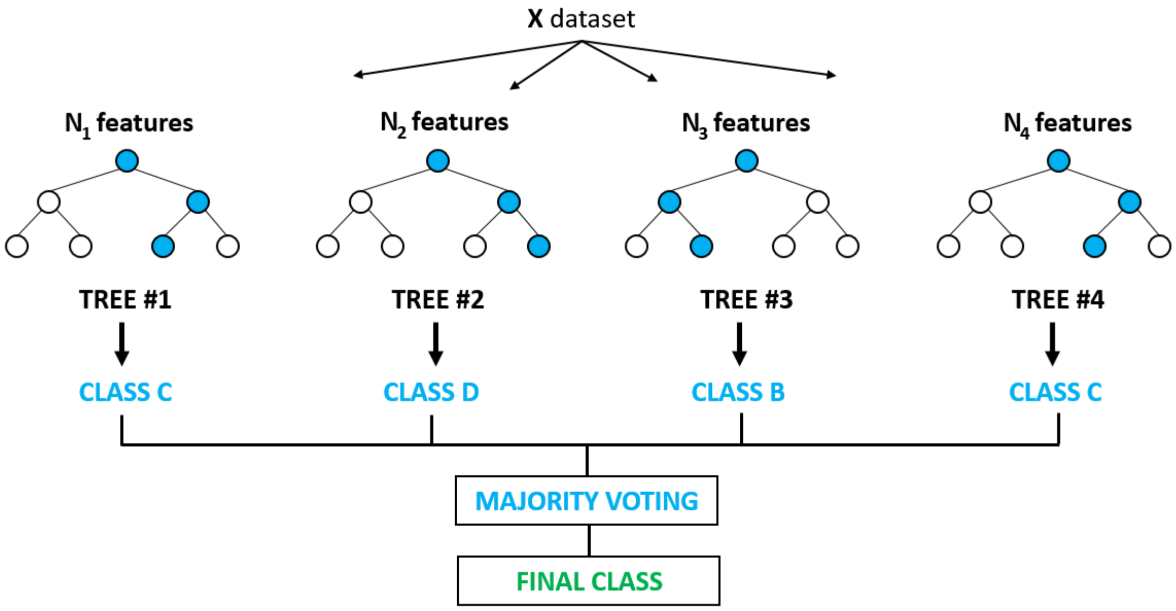


Figure 4.2.8 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

4.2.9 ANN – Artificial Neural Network

We build ANN using Vector representations in one model. In another model, we use all the linguistic features – 3 linguistic features.

Sentiment – notion of tone of the writer

Punctuation count – total number of punctuations

Readability – ease of understanding

Below, basic notion behind ANNs has been briefly discussed.

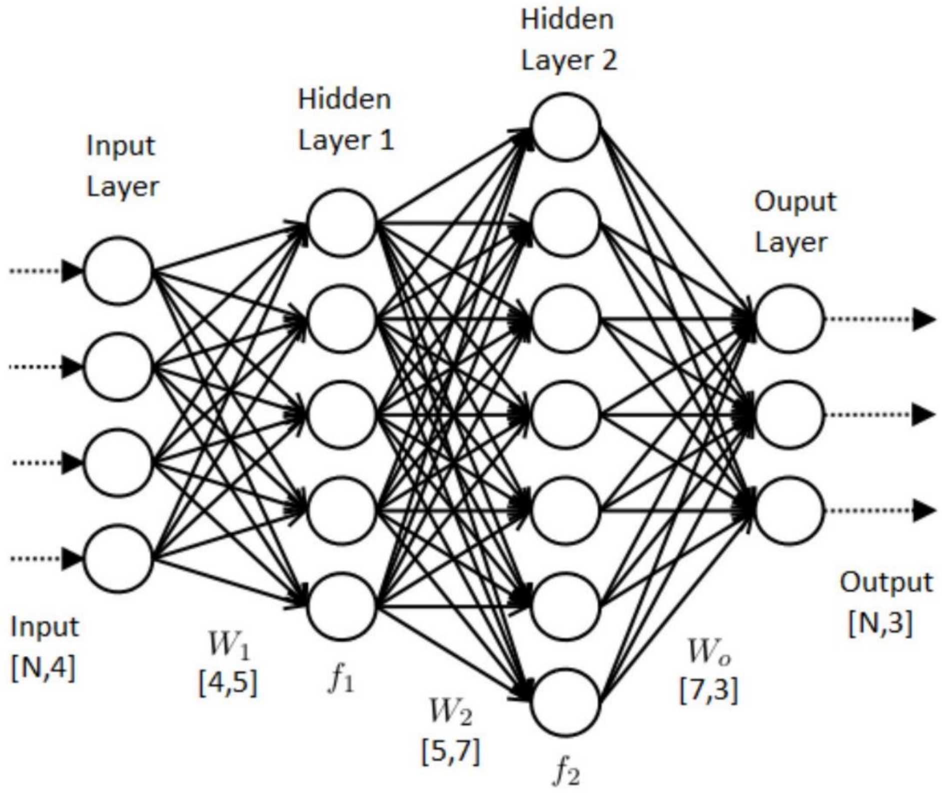


Figure 4.2.9 ANN

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

In ANN implementations, the "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times. The original goal of the ANN approach was to solve problems in the same way that a human brain would. But over time, attention moved to performing specific tasks, leading to deviations from biology. ANNs have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis, and even in activities that have traditionally been considered as reserved to humans, like painting

A complete, production-quality classifier will incorporate many different features beyond the vectors corresponding to the words in the text. For fake news detection, we can add as features the source of the news, including any associated URLs, the topic (e.g., science, politics, sports, etc.), publishing medium (blog, print, social media), country or geographic region of origin, publication year, as well as linguistic features not exploited in this exercise use of capitalization, fraction of words that are proper nouns (using gazetteers), and others. Besides, we can also aggregate the well-performed classifiers to achieve better accuracy. For example, using bootstrap aggregating for the SVM model to get better prediction result. We would also explore other classification models in ML. An ambitious work would be to search the news on the Internet and compare the search results with the original news. Since the search result is usually reliable, this method should be more accurate, but also involves natural language understanding because the search results will not be exactly the same as the original news. So, we will need to compare the meaning of two contents and decide whether they mean the same thing.

This chapter provided details about the implementation done It also elaborated upon further work in present implementation.

In the next chapter we will discuss the results of the present implementation.

Chapter5. RESULTS AND DISCUSSIONS

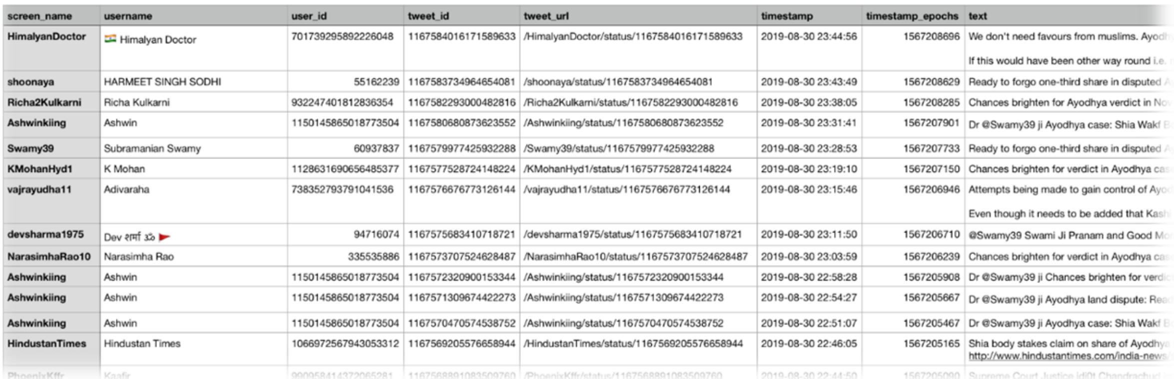
Here we discuss the details of our collected dataset as of now. We also compare the performances of the ML models we have utilized up till now

5.1 Data

We have more than lakh records for twitter with attributes as:

With the twitter scrapper python package, it is very easy to retrieve large amounts of data corresponding to a particular query.

We also have crawlers for as many as 8 newspapers, and the combined data from these contains 5k+ records. All these sites have different interfaces, no generic scrapper works for all of them. The bigger problem is that most classification approaches are supervised so we need prior dataset to train our model but we see that obtaining a reliable fake news dataset is very time-consuming process.



screen_name	username	user_id	tweet_id	tweet_url	timestamp	timestamp_epochs	text
HimalyanDoctor	Himalyan Doctor	701739295892226048	1167584016171589633	/HimalyanDoctor/status/1167584016171589633	2019-08-30 23:44:56	1567208696	We don't need favours from muslims. Ayodh If this would have been other way round i.e.
shoonaya	HARMEET SINGH SODHI	55162239	1167583734964654081	/shoonaya/status/1167583734964654081	2019-08-30 23:43:49	1567208629	Ready to forgo one-third share in disputed
Richa2Kulkarni	Richa Kulkarni	932247401812836354	1167582293000482816	/Richa2Kulkarni/status/1167582293000482816	2019-08-30 23:38:05	1567208285	Chances brighten for Ayodhya verdict in No
Ashwinkiing	Ashwin	1150145865018773504	1167580680873623552	/Ashwinkiing/status/1167580680873623552	2019-08-30 23:31:41	1567207901	Dr @Swamy39 ji Ayodhya case: Shia Wakf
Swamy39	Subramanian Swamy	60937837	1167579977425932288	/Swamy39/status/1167579977425932288	2019-08-30 23:28:53	1567207733	Ready to forgo one-third share in disputed
KMohanHyd1	K Mohan	1128631690656485377	1167577528724148224	/KMohanHyd1/status/1167577528724148224	2019-08-30 23:19:10	1567207150	Chances brighten for verdict in Ayodhya ca
vajrayudha11	Adivaraha	738352793791041536	1167576676773126144	/vajrayudha11/status/1167576676773126144	2019-08-30 23:15:46	1567206946	Attempts being made to gain control of Ayo Even though it needs to be added that Kash
devsharma1975	Dev शर्मा ॐ	94716074	1167575683410718721	/devsharma1975/status/1167575683410718721	2019-08-30 23:11:50	1567206710	@Swamy39 Swami Ji Pranam and Good Mo
NarasimhaRao10	Narasimha Rao	335535886	1167573707524628487	/NarasimhaRao10/status/1167573707524628487	2019-08-30 23:03:59	1567206239	Chances brighten for verdict in Ayodhya ca
Ashwinkiing	Ashwin	1150145865018773504	1167572320900153344	/Ashwinkiing/status/1167572320900153344	2019-08-30 22:58:28	1567205908	Dr @Swamy39 ji Chances brighten for verdi
Ashwinkiing	Ashwin	1150145865018773504	1167571309674422273	/Ashwinkiing/status/1167571309674422273	2019-08-30 22:54:27	1567205667	Dr @Swamy39 ji Ayodhya land dispute: Rea
Ashwinkiing	Ashwin	1150145865018773504	1167570470574538752	/Ashwinkiing/status/1167570470574538752	2019-08-30 22:51:07	1567205467	Dr @Swamy39 ji Ayodhya case: Shia Wakf
HindustanTimes	Hindustan Times	1066972567943053312	1167569205576658944	/HindustanTimes/status/1167569205576658944	2019-08-30 22:46:05	1567205165	Shia body stakes claim on share of Ayodhy http://www.hindustantimes.com/india-news
PhoenixKffr	Kaafir	990958414372065281	1167568891083509760	/PhoenixKffr/status/1167568891083509760	2019-08-30 22:44:50	1567205090	Supreme Court Justice idi0t Chandrachud

Figure 5.0.1: Twitter data illustration

combined_csv

id	title	author	text	label
	Sexism in chemistry, FDA nomination and a long-lost ungulate	Abc	The latest science news, in brief.	1
	Central GST Delhi-West Commissionerate unearthed racket of issuance of fake invoices	Abc	Central GST Delhi-West Commissionerate has unearthed racket of issuance of fake invoices without actual supply of goods and services by M/s Royal Sales India and 27 other dummy companies. Two persons have been ar	1
	Blog: Let opposition parties back into Kashmir	Abc	Whether you agree or disagree with the idea of getting 23 MPs from the European Union to visit Kashmir, the visit opened up key questions. Was it a limited public relations photo-op or the beginning...	1
	Police cautions people against fake news ahead of Ayodhya verdict	Abc	Respect Supreme Court's judgment, SP tells people	1
	Govt. plans to split State: Ambedkar	Sahi	'BJP using President's Rule like in J&K'	1
	Police crack whip against those sticking threatening posters in Valley, says IGP Pani	Sahi	The IGP said several persons have been arrested from different parts of the Valley including Srinagar.	1
	Congress stages walkout in Lok Sabha	Sahi	Speaker disallows Manish Tewary's reference to PMO on electoral bonds issue.	1
	UK political morass deepens as Farage threatens Johnson	Sahi	Leader of Brexit Party wants UK PM Johnson to join hands with him, as suggested by US President Trump.	1
	Over 5,000 people arrested since August 4 in Kashmir: Home Ministry	Sahi	These arrests were preventive in nature and a majority of those arrested have been released since. Currently, 609 people, 218 of whom are alleged stone pelters, continue to be under detention, Ministry of Home Affairs told F	1
	SC reserves verdict on pleas challenging curbs in J&K	Sahi	The Supreme Court reserved on Wednesday its verdict on a batch of pleas including that of Congress leader Ghulam Nabi Azad challenging the restriction imposed in the erstwhile state of Jammu and Kashm	1
	Ayodhya on tenterhooks: Some shift out, others stocking up on ration	Sahi	In Ayodhya these days, the air is thick with apprehension and anxiety. With the temple verdict expected any day now, the town's residents are making whatever preparations they can — some are hoarding food and essentials	1
	Sexism in chemistry, FDA nomination and a long-lost ungulate	Sahi	The latest science news, in brief.	1
	Ayodhya: How a religious issue became a political hot potato	Sahi	Given its potential for communal polarisation, it could not ever be detached from politics	1
	Soon, take chopper between Mumbai, Pune, Shirdi	Guy	Inaugural fares for Mumbai-Pune Rs 19,900, Mumbai-Shirdi at Rs 21,900 & Pune-Shirdi at Rs 18,900.	1
	Finance Commission likely to get extension for working on devolution to UTs of J&K, Ladakh	Guy	According to the Act, the distribution of taxes suggested by the 15th Finance Commission to the state of Jammu and Kashmir will have to be apportioned to the Union Territories of J&K and Ladakh. The award made by the F	1
	Love of cash hinders India's move to digital economy	Guy	India's dependency on cash may slow the country's transition to digital payments despite large numbers of internet and mobile phone users. Sudhir Shinde, a farmer from Satara district in India's western state of Maharashtr	1
	An update on our political ads policy	bhGuy	We're proud that people around the world use Google to find relevant information about elections and that candidates use Google and search ads to raise small-dollar donations that help fund their campaigns. We're also co	1
	BPO firms left in the lurch over input tax refund	bhGuy	The Maharashtra Appellate Authority for Advance Ruling (AAAR) had held in a ruling in February that back-office support services did not qualify as "export of service" and were in the nature of arranging or facilitating supply	1
	1045 page Ram Mandir Verdict to be included in JNU syllabus to make students stay longer	bhGuy	The #AyodhyaVerdict is finally out. The established practice is to specify the name of the judge who has authored the judgment on behalf of a bench. The reason behind the decision to not mention the name of the author of t	0
	Congress wants a 'grand temple' in Ayodhya: Pilot	bhGuy	India News: JAIPUR: The Congress wants a "grand temple" to be built in Ayodhya, Rajasthan deputy chief minister Sachin Pilot said on Friday, days after the Suprem.	1
	BJP leader sees Pak. hand behind pollution	bhGuy	'New ploy of releasing toxic gases'	1
	Meelath Committee accepts SC verdict on Ayodhya issue	bhGuy	The members of the Chennai City Meelath Committee on Sunday said that it accepted the Supreme Court decision on the Babri Masjid-Ram Janmabhoomi dispute and called for all the communities to live in u	1
	Donald Trump Says he Can Mediate On Ram Mandir Issue If Both Parties Agree	bhGuy	Soon after US President Donald Trump offered to mediate between India and Pakistan on Kashmir issue a major part of the issue got resolved. With the abrogation of Article 370, India government has made J&K a Union Terr	0
	Industries saying they are out of distress: FM	bhGuy	India Business News: In an exclusive interview with Times of India, finance minister Nirmala Sitharaman talks about how government's measures are starting to show results.	1
	India caucus co-chair statement to U.S. Congressional Record supports Modi	bhGuy	Indian officials lobby Capitol Hill after last week's hearing on Kashmir	1
	Anger over India's diplomat calling for 'Israel model' in Kashmir	Cos	In video posted by filmmaker, India's Consul General to US seems to advocate Israel-like settlements of Kashmiri Hindus.	1
	EU MPs' day out: Shikara ride amid protests, briefing on J&K	Cos	In the document titled 'Briefing notes and background on Kashmir', the government provided the visiting EU MPs a condensed account of events that unfolded in the Valley over the past three months.	1
	Has MTech fee hike led to a decline in GATE 2020 registrations?	Cos	GATE 2020: Despite the lockdown in the Valley following the scrapping of Article 370, over 10,000 from the region have applied for GATE 2020. As per the data provided by the GATE 2020 office, a total of 5464 candidates ap	1
	T.N. Seshan redefined very grammar of electoral process: ECI	Cos	The ECI, including CEC Sunil Arora and Election Commissioner Ashok Lavasa, held a condolence meeting for Seshan at its headquarters here on Monday morning	1
	SECURE SYNOPSIS: 11 NOVEMBER 2019	Cosvg	SECURE SYNOPSIS: 11 NOVEMBER 2019 NOTE: Please remember that following 'answers' are NOT 'model answers'. They are NOT synopsis too if we go by definition of the term. What we are providing is content that bot	1
	Facebook starts 'Mark Yourself Safe' feature for Maharashtra MLAs	Cos	Mumbai. Maharashtra politics has been as entertaining as a Bollywood thriller. Each day comes up with new surprises and it is a never-ending story. Abbas Mastan have in fact bought the rights of making the movie on Maha	0
	RCEP agreement: Can ill-afford to be dumping ground for products, says Sonia Gandhi	Cos	Sonia Gandhi was speaking to top leaders who were meeting to review preparations for a ten-day nationwide agitation to be launched from November 5.	1
	Bhandari makes a case for govt intervention in stressed NBFCs	Cos	Until NBFCs revive, rural incomes won't revive in a big way, says HSBC Chief India Economist	1
	Full text of Ayodhya verdict	Cos	The Supreme Court on Saturday cleared the way for the construction of a Ram Temple at the disputed site at Ayodhya, and directed the Centre to allot a 5-acre plot to the Sunni Waqf Board for building	1
	Exemptions for Development to MSMEs	Cos	Under Goods and Services Tax (GST) Act, Micro & Small units with turnover up to Rs. 1.5 crore are allowed to avail benefits under the composition scheme. Ministry of Micro, Small and Medium Enterprises (MSME) has launc	1

Figure 5.0.2 News Website Scrapped Data Illustration

5.2 Machine Learning

As of now, we have only experimented with two ML models – SVM and Naive Bayes, he results of both are depicted below.

5.2.1 Doc2Vec

	The ruling paves the way for Hindus to build a temple where the Babri Mosque once stood, a decision that raised fears
	Hindus and Muslims have long sparred over a few barren acres in the city of Ayodhya. But since Prime Minister Narend
	LEAD: At the end of a pilgrims' hilly path, where temple monkeys sun themselves and peddlers hustle their marigold ga
	A long-awaited decision on control of India's most disputed religious site splits the land into three portions to be divide
	Hindu nationalists rose to electoral significance from the debris of the mosque they demolished in 1992.NEW DELHI —
	A mix of hope and fear defined his first five years as prime minister. Both supporters and critics wonder whether a defir
	An Indian court ruled Thursday that a disputed holy site in Ayodhya, India, be divided between Hindus and Muslims.An
	Police fire tear gas at hundreds of demonstrators in New Delhi who are protesting attack on disputed religious site, tem
	Indian Prime Min Atal Bihari Vajpayee insists that Hindus will build contested temple in Ayodhya, at site where 16th-cen
	Court in northern India orders archaeologists to begin excavating holy site in Ayodhya to determine whether Hindu tem
	Hundreds of thousands of lamps illuminated the northern city of Ayodhya for Diwali, casting a glowing light over the cit
	Temple that Hindu hard-liners seek to build on site of razed 16th-century mosque in Ayodhya continues to create probl
	Three bombs kill 11 people and injure 54 on trains in southern India, raising fears that campaign for general election in
	Five assailants were killed at Ayodhya during a firefight and the country went on high alert in anticipation of potential co
	Thousands of police officers and paramilitary troops armed with tear gas, riot sticks and guns fought off determined ba
	In a rare display of political unity, India's bitterly divided governing and opposition parties today jointly appealed to Hin
	A brazen attack on India's best-known tinderbox of Hindu-Muslim strife, the heavily fortified Hindu temple compound ir
	Leaders of a campaign to build a Hindu temple near a mosque in the northern town of Ayodhya said today that they h
	LEAD: Hundreds of thousands of Hindu militants gathered today near a shrine disputed by Hindus and Muslims to asse
	A mob of 5,000 to 10,000 militant Hindus tried again today to storm a disputed mosque in the holy city of Ayodhya on
	Indian Paliament is disrupted for fifth day in row by opposition demand for resignations of three cabinet ministers from

Figure 5.2.1.1 Initial input

```
x = constructLabeledSentences(data['text'])
y = data['label'].values

text_model = Doc2Vec(min_count=1, window=5, vector_size=vector_dimension, sample=1e-4, negative=5, workers=7, epochs=10,
                     seed=1)
text_model.build_vocab(x)
text_model.train(x, total_examples=text_model.corpus_count, epochs=text_model.iter)

train_size = int(0.8 * len(x))
test_size = len(x) - train_size

text_train_arrays = np.zeros((train_size, vector_dimension))
text_test_arrays = np.zeros((test_size, vector_dimension))
train_labels = np.zeros(train_size)
test_labels = np.zeros(test_size)

for i in range(train_size):
    text_train_arrays[i] = text_model.docvecs['Text_' + str(i)]
    train_labels[i] = y[i]
```

Figure 5.2.1.2 Transformation Core

```
print(xte[82].shape)
print(xte[82])
print(xte)
```

```
(300,)
[ 1.55134918e-03  1.29717879e-03  1.60206028e-03 -1.11021881e-03
  5.21221838e-04 -4.98319976e-04 -3.46861081e-04 -1.49420768e-04
 -7.23557721e-04 -1.19458931e-03 -7.98369176e-04  1.14441616e-03
  5.29741053e-04  1.51328486e-03 -1.53085170e-03 -7.76541536e-04
 -5.16988803e-04  2.92461395e-04 -2.50334357e-04 -3.31959862e-04
 -6.16274599e-04  4.21455188e-04  1.00360357e-03  1.09911885e-03
 -1.32122519e-03  3.02799686e-04  1.63728037e-04 -2.83378176e-04
 -8.84639885e-05 -2.68198608e-04 -8.13012768e-04 -1.12096581e-03
  1.08408218e-03  7.99731293e-04  8.70218850e-04  9.58594144e-04
 -5.90207401e-06  4.01583238e-04  1.65923708e-03  1.42471900e-03
```

Figure 5.2.1.3 Word Embeddings output

5.2.2 TFIDF

```
In [20]: tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)
         tfidf_train=tfidf_vectorizer.fit_transform(x_train)
         tfidf_test=tfidf_vectorizer.transform(x_test)

In [21]: tfidf_train

Out[21]: <14628x151206 sparse matrix of type '<class 'numpy.float64'>'
                 with 3991333 stored elements in Compressed Sparse Row format>

In [24]: tfidf_train[0]

Out[24]: <1x151206 sparse matrix of type '<class 'numpy.float64'>'
                 with 215 stored elements in Compressed Sparse Row format>
```

Figure 5.2.2 TFIDF

5.2.3 ML models

5.2.3.1 SVM

INPUT	**ACCURACY**
TFIDF	**95.14 %**
Doc2Vec	**91.58 %**
Sentiment Score	**61 %**
Punctuation Count	**64 %**
Readability	**65 %**
All 3 Linguistic Features	**69 %**

Table 5.2.3.1 SVM

5.2.3.2 KNN

INPUT	**ACCURACY**
TFIDF	**86.03 %**
Doc2Vec	**86.77 %**
Sentiment Score	**57 %**
Punctuation Count	**58 %**
Readability	**60 %**
All 3 Linguistic Features	**68 %**

Table 5.2.3.2 KNN

5.2.3.3 Logistic Regression

INPUT	**ACCURACY**
TFIDF	**95.13 %**
Doc2Vec	**90.10 %**
Sentiment Score	**57 %**
Punctuation Count	**57 %**
Readability	**67 %**
All 3 Linguistic Features	**67 %**

Table 5.2.3.3 Logistic Regression

5.2.3.4 Naive Bayes

INPUT	ACCURACY
TFIDF	78.18 %
Doc2Vec	74.32 %
Sentiment Score	57 %
Punctuation Count	57 %
Readability	61 %
All 3 Linguistic Features	61 %

Table 5.2.3.4 Naïve Bayes

5.2.3.5 Decision Tree

INPUT	ACCURACY
TFIDF	90.43%
Doc2Vec	70.74 %
Sentiment Score	59%
Punctuation Count	63%
Readability	65%
All 3 Linguistic Features	63%

Table 5.2.3.5 Decision Tree

5.2.3.6 Random Forest

INPUT	ACCURACY
TFIDF	90.00 %
Doc2Vec	88.79 %
Sentiment Score	59 %
Punctuation Count	62 %
Readability	65 %
All 3 Linguistic Features	70 %

Table 5.2.3.6 Random Forest

5.2.3.7 ANN

INPUT	**ACCURACY**
TFIDF	**95.6 %**
Doc2Vec	**92.62 %**
All 3 Linguistic Features	**71.53 %**

Table 5.2.3.7 ANN

In this chapter we discussed the results we obtained from classical ML models and ANN. We also discussed the amount and sources of data we have been able to extract.

Chapter: 6 CONCLUSIONS

6.1 Data's All folks!

To make a robust ML model, the foremost requirement is reliable and sufficient amount of data. It is difficult and time consuming to build a big labelled dataset to train the classification models efficiently. With changing nature of fake news, other challenge is to make his data dynamic that updates automatically keeping at par with latest trend of fake news.

6.1.1 Artificial Data

Another way to deal shortage of data is to create and use artificial data provided that it does not compromises with the performance of the resulting ML models. We can explore various areas for generating fake news. For eg: changing few punctuations, stop words in already fake news. We can also explore GANs for generating fake news.

6.1.2 Dynamic Data and Robustness

Database once formed and used to train ML model keeps on aging with time. Sooner or later it will become stale and will not be best suited to detect fake news in the modern times. We need to formalize some strategies to make the database dynamic that keeps in par with the latest trends in this area.

6.2 Machine Learning

We aim to explore more classification algorithms and finally experiment with GANs and Deep Learning.

6.2.1 Other elementary methods

Classification is a classical application of ML. A number of algorithms have been proposed to classify a group of items into categories. Eg: e.g. Decision tree, logistic regression, K nearest neighbours. Different algorithms yield different results based on the circumstances

6.2.2 Feature Engineering – more features

We can analyze fake news differently with different measure similarities e.g. Location, Time, author and Quality. We can detect whether the same news published by other media agencies or not, We can check the location of the news. Maybe a news has a higher probability of being fake, if it is generated somewhere else and not at the location they deal with (e.g. News about Captain Amrinder Singh has its origin in Israel), we can check news quality wise it is more probable that fake news do not have mentioned their sources, simply claim something, while for real news the source is mention and also we can check the time of the news as whether the same news appears in other media or sourced if it is repeated more often in the beginning, because they are interesting, and become recognized as fake with the time, which reduces the repetition or they are deleted from some websites. At this stage we don't have definitive solution but after detailed literature review we can say that it's true that producing more reports with more facts can be useful for helping us to make such decisions and find technical solutions in fake news detection.

6.2.3 Ensemble Techniques

Ensemble techniques can be promising. Random forest has been a successful ensemble combining decision tress as the underneath algorithm. We should experiment with ensemble techniques using different strategies of boosting and parallezing various algforithms to tackle Fake News.

References:

1. Wikipedia

2. Research papers

- Traore, Issa & Saad, Sherif. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. 127-138. 10.1007/978-3-319-69155-8_9.

-Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2018. Combating Fake News: A Survey on Identification and Mitigation Techniques. ACM Trans. Intell. Syst. Technol. 37, 4, Article 111 (August 2018), 41 pages. https://doi.org/10.1145/1122445.1122456

-Torabi Asr, Fatemeh Taboada, Maite http://orcid.org/0000-0002-6750-8891,Big Data & Society January–June 2019: 1–14 Article reuse guidelines: sagepub.com/journals-permissions DOI: 10.1177/2053951719843310 journals.sagepub.com/home/bds

-M. Singh et al. (Eds.): ICACDS 2019, CCIS 1046, pp. 420–430, 2019.

3. Udemy ML Course, A-Z Machine Learning

4. skymind.ai

5. Coursera ML Course BY Andrew NJ

6. expertsystem.com

7. datameer.com

8. becominghuman.ai

9. data-flair.training

10. techcrunch.com

11. leadingindia.ai

12. www.python.org

13. Stack Overflow

14. simpleprogramming.com

15. towardsdatascience.com

16. medium.com

17. developers.google.com

18. analyticsvindhya.com