MINI PROJECT

(2021-22)

"Fake News Detection Using Machine Learning"

Project Report

Institute of Engineering & Technology

Submitted By -

Ojasva Saxena (191500520)

Pranjal Bansal (191500569)

Shriyanshi Patwa (191500790)

Soniya Sahu(191500820)

Under the Supervision Of

Mr. Amir khan

Technical Trainer

Department of Computer Engineering & Applications

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Declaration

I/we hereby declare that the work which is being presented in the Bachelor of technology. Project **"Fake News Detection Using Machine Learning"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr. Amir Khan, Technical Trainer, Dept. of CEA,GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: Ojasva Saxena
Name of Candidate: Ojasva Saxena
University Roll No.: 191500520

Sign: Pranjal Bansal
Name of Candidate: Pranjal Bansal
University Roll No.:191500569

Sign: Shriyanshi Patwa
Name of Candidate: Shriyanshi Patwa
University Roll no.-191500790

Sign: Soniya Sahu
Name of Candidate: Soniya Sahu
University Roll No.:191500820

Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

Certificate

This is to certify that the project entitled "**Fake News Detection Using Machine Learning"**", carried out in Mini Project – I Lab, is a bonafide work by Ojasva Saxena, Pranjal Bansal, Soniya Sahu, Shriyanshi Patwa is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Mr. Aamir Khan
Date: 15-11-21

Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mr Aamir Khan, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign:*Pranjal Bansal*
Name of Candidate: Pranjal Bansal
University Roll No.:191500569

Sign: *Ojasva Saxena*
Name of Candidate: Ojasva Saxena
University Roll No.:191500520

Sign: *Shriyanshi Patwa*
Name of Candidate: Shriyanshi Patwa
University Roll No.:191500790

Sign: *Soniya Sahu*
Name of Candidate: Soniya Sahu
University Roll No.:191500820

ABSTRACT

In this project we are creating a platform regarding the credibility of the news that is being spread all around the places. It is basically a quick access towards the authenticity. The current large web of social media platforms have increased the probability of fake news that are being circulated in the messages and mails of the users, in order to check the credibility of the particular point this platform is quite helpful. The project uses the five widely used machine learning methods: Long Short Term Memory (LSTM), Random Forest (random tree), Random Forest (decision tree), Decision Tree.

Any news whose authenticity and source cannot be validated by the reader is termed as a fake news. The issue of fake news has become a serious problem in India because of high digital illiteracy and low digital penetration. Like any other social phenomenon fake news also has its own pros and cons. The cons are discussed in the next section. There sources many sources of different news. Some sources are authoritative for example the government websites and others are licensed. In both of these cases the identity of the source can be easily identified. The problem occurs when the source of a news cannot be determined by the authorities and the social media comes into the frame. Social media is a decentralized source of information with minimal credibility.

CONTENTS

CHAPTER-1

INTRODUCTION

1.1 CONTEXT

This Machine Learning Application “Fake News Detection” has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by Mr. Amir Khan. This project has been completed approximately one month and has been executed in modules, meetings have been organised to check the progress of the work and for instructions and guidelines.

1.2 MOTIVATION

In the recent years as we have seen that the social media platforms have increased rapidly and along with that the amount of fake news has been booming up as well. In order to keep the users updated and to help them from being influenced by the fake news this platform is a small step towards the retention of such acts.

The news that are been spread can be on motive or can also be a mistake from the other end. 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. In the past presidential election, the American people were overwhelmed with the proliferation of “fake news” articles that altered the narrative (and perhaps the results) of the election. The articles and social media posts featured bombastic headlines and made outrageous claims regarding the candidates.

1.3 OBJECTIVE

This project will contribute to the start of a new revolution against one of the most prevalent hazard i.e. spread of the Fake News. It will serve as root and branch eradication of the same. This project will help to create a next level of awareness and make the citizens more responsible. This project will help the people of a nation to take meaningful and informed decisions.

1.4 EXISTING DATASET FOR THIS SYSTEM

The lack of manually labeled fake news datasets is certainly a bottleneck for advancing computationally intensive, text-based models that cover a wide array of topics. The dataset for the fake news challenge does not suit our purpose due to the fact that it contains the ground truth regarding the relationships between texts but not whether or not those texts are actually true or false statements. For our purpose, we need a set of news articles that is directly classified into categories of news types (i.e. real vs. fake or real vs parody vs. clickbait vs. propaganda). For more simple and common NLP classification tasks, such as sentiment analysis, there is an abundance of labeled data from a variety of sources including Twitter, Amazon Reviews, and IMDb Reviews. Unfortunately, the same is not true for finding labeled articles of fake and real news. This presents a challenge to researchers and data scientists who want to explore the topic by implementing supervised machine learning techniques. I have researched the available datasets for sentence-level classification and ways to combine datasets to create full sets with positive and negative examples for document-level classification.

1.5 SOURCES

The source of our project (including all the project work, documentations and presentations) will is available at the following link

https://github.com/pranjalbansal200/imposter_news_detection

CHAPTER -2
SOFTWARE REQUIREMENT ANALYSIS

2.1 WHAT IS FAKE NEWS?

Fake news has quickly become a society problem, being used to propagate false or rumour information in order to change peoples behaviour. It has been shown that propagation of fake news has had a non-negligible influence of 2016 US presidential elections. A few facts on fake news in the United States:

- 62% of US citizens get their news for social medias.
- Fake news had more share on Facebook than mainstream news.

The first is characterization or what is fake news and the second is detection. In order to build detection models, it is need to start by characterization, indeed, it is need to understand what is fake news before trying to detect them.

Fake news definition is made of two parts: authenticity and intent. Authenticity means that fake news content false information that can be verified as such, which means that conspiracy theory is not included in fake news as there are difficult to be proven true or false in most cases. The second part, intent, means that the false information has been written with the goal of misleading the reader.

2.2 PROBLEM STATEMENT

News consumption is a double-edged sword. On the one hand, its low cost, easy access, and rapid dissemination of information lead people to seek out and consume news.It enables the wide spread of “fake news”, i.e., low quality news with intentionally false information. The extensive spread of fake news has the potential

for extremely negative impacts on individuals and society. Therefore, fake news detection has recently become an emerging research that is attracting tremendous attention.First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content. To develop a FAKE NEWS DETECTION system using natural language processing and its accuracy will be tested using machine learning algorithms. The algorithm must be able to detect fake news in a given scenario.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

SYSTEM CONFIGURATION

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

Hardware Requirement

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

Software Requirement

- Python 3.5 in Google Colab is used for data pre-processing, model training and prediction.

- Operating System: windows 7 and above or Linux based OS or MAC OS.

CHAPTER- 3

SYSTEM DESIGN

3.1 PROPOSED SYSTEM

The proposed system when subjected to a scenario of a set of news articles , the new articles are categorized as true or fake by the existing data available . This prediction is done by using the relationship between the words used in the article with one another. The proposed system contains a Word2Vec model for finding the relationship between the words and with the obtained information of the existing relations , the new articles are categorized into fake and real news.

3.2 SYSTEM ARCHITECTURE

The Proposed system's architecture is shown I the below. In this, we are introducing a new tool Doc2Vec to predict fake news and tell us the accuracy. . User must obtain the predicted output in a accurate manner and time-efficient manner. The project is implemented using python and other machine learning algorithms, since the resources are available

There are three modules namely

- Extracting the data
- Data pre-processing
- Prediction.

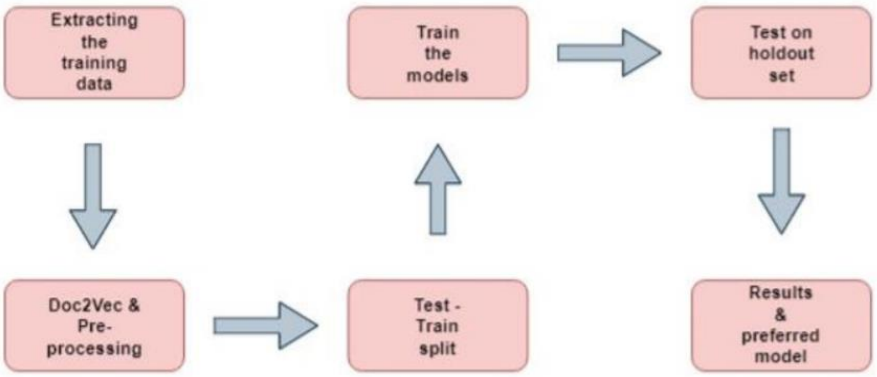


Figure 5.1 Architecture of FDM

3.2.1 CLASS DIAGRAM

The Class diagram has the ability to make the developer to understand the workflow in a better conceptual manner. It is also one of the essential block of the OOPS (Object-Oriented Modelling) concept. It is widely used for the common conceptual modelling of designing and system architecture or making a systematic approach for development of applications. It is also useful for briefing and modelling the translation of the study design model to the code. Class diagrams are also used for the data modeling. The main elements, interactions in the application, and the classes to be programmed are defined and represented in the class diagram by different classes.

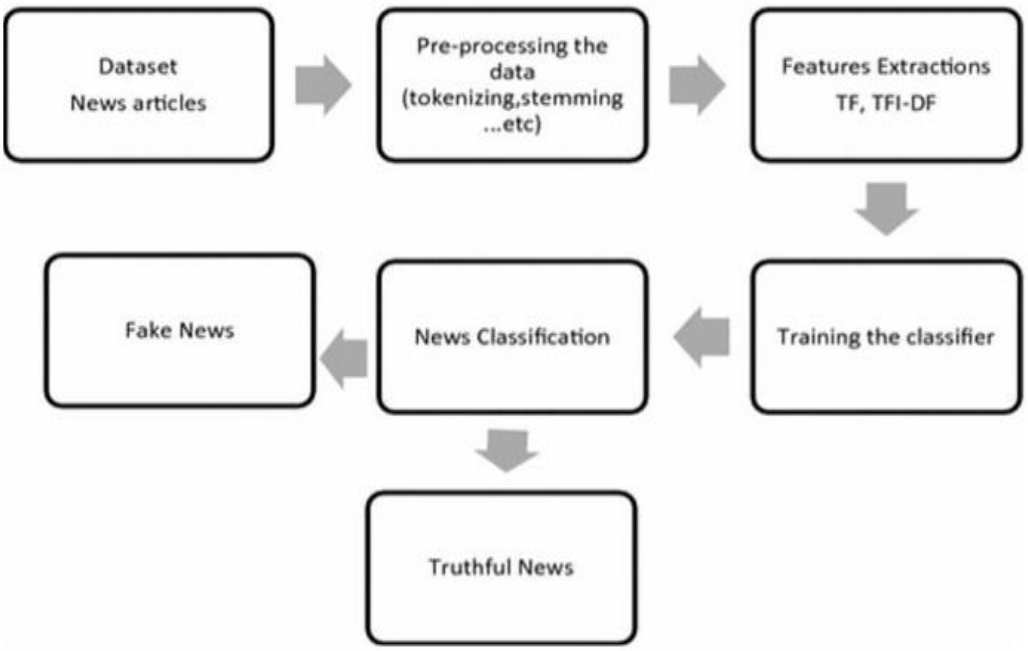
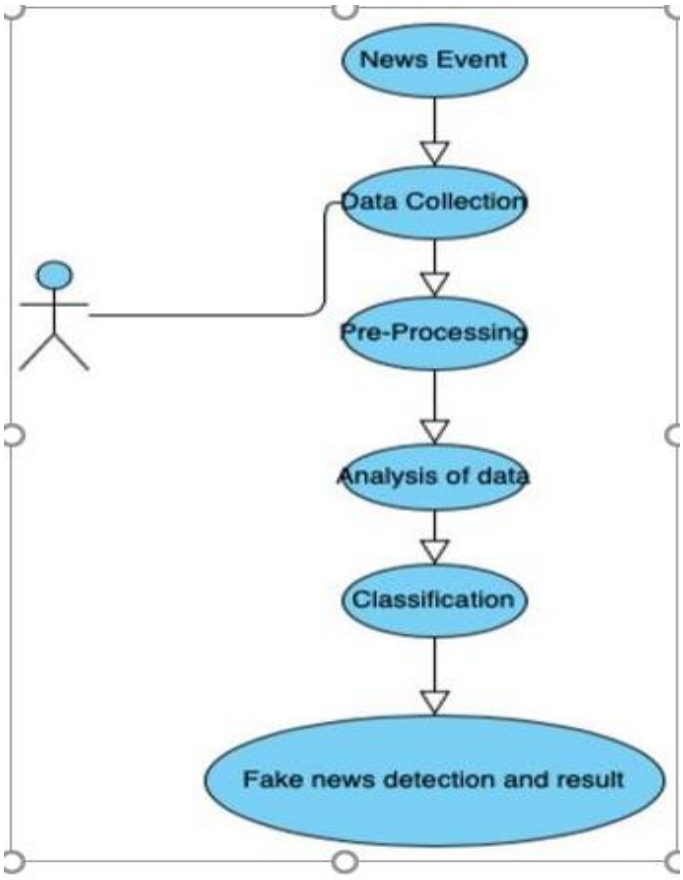


Figure 5.2 Class Diagram of FDM

3.2.2 USE CASE DIAGRAM

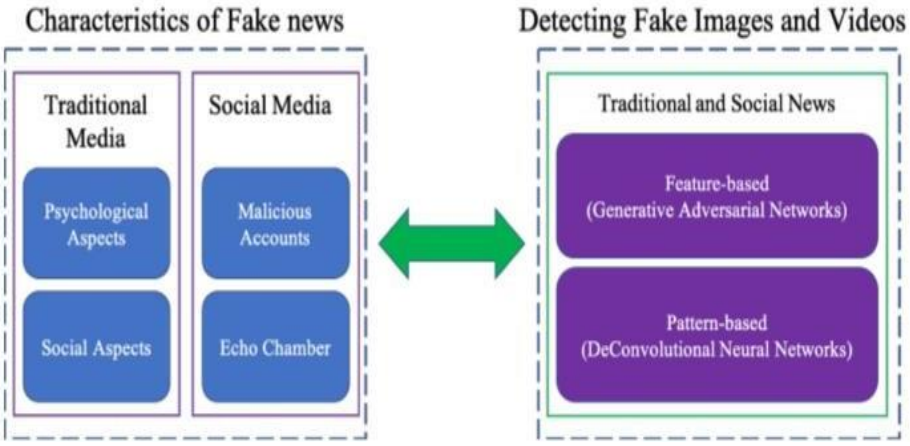
The Use Case diagram is one of the best and the simple diagram or a design to show to relationship between the user's interactions. It also tells about the relationship between the userand the various use cases. The Use Case diagram can find the different types of the users in the system and the cases, which are used in the other structure diagrams. With the help of Use Casediagram higher-level view of the proposed system can be depicted.

They represent us the workflow of what system does in a simple graphical representation. Because of their simple design, it is serving as good communication tool for stakeholders.



3.2.3 SEQUENCE DIAGRAM

Sequence diagrams are made usually to make the user and stakeholders to understand the interaction of objects and how they are arranged in the sequence manner. The Objects and the classes are involved it and plays the important role. The sequence of the news collected by the object and defined in various classes which carry the functionality feature of the scenario are depicted here in the sequence diagram. The Use Case realization are shown in the Logical view of the system model development and are associated with the sequence diagrams. They are alsocalled as Event diagrams .



CHAPTER-4
TECHNOLOGY USED

4.1 MACHINE LEARNING

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms **for** building mathematical models and making predictions using historical data or information**.** Currently, it is being used for various tasks such as image recognition**,** speech recognition**,** email filtering**,** Facebook auto-tagging**,** recommender system**,** and many more.

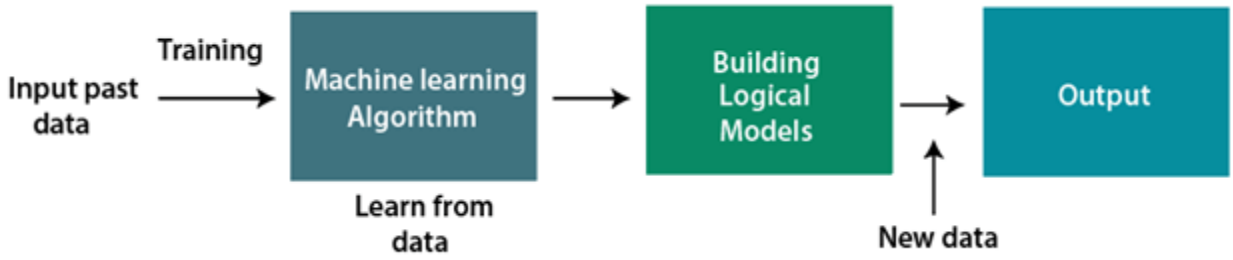
In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role **of** Machine Learning**.**

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel **in** 1959. We can define it in a summarized way as:

How does Machine Learning work

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in self-driving cars**,** cyber fraud detection**,** face recognition**,** and friend suggestion by Facebook**,** etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

4.2 TOOLS AND LANGUAGES

TOOLS

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Getting Up and Running With Jupyter Notebook

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

Installation

If so, then you can use a handy tool that comes with Python called **pip** to install Jupyter Notebook like this:

```
$ pip install jupyter
```

The Jupyter Notebook has several menus that you can use to interact with your Notebook. The menu runs along the top of the Notebook just like menus do in other applications. Here is a list of the current menus:

- *File*
- *Edit*
- *View*
- *Insert*

- *Cell*
- *Kernel*
- *Widgets*
- *Help*

LANGUAGES

Python (programming language)
General-purpose programming language

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected.

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. For example, a Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program.

Applications

The Python Package Index (PyPI) hosts thousands of third-party modules for Python. Python's standard library and the community-contributed modules allow for endless possibilities.

- Web and Internet Development
- Database Access
- Desktop GUIs
- Scientific & Numeric
- Education
- Network Programming

CHAPTER -5
IMPLEMENTATION

Step to be followed:

Step 1: Start

Step 2: Input is collected from various sources and prepare a dataset.

Step 3: Preprocessing of data is done and dataset is divided into 2 parts training and testing data.

Step 4: Count vectorization technique is used to convert the train data into numericals.

Step 5: Different algorithms are used to build the predictive model using the train data .

Step 6: Confusion matrix is obtained .

Step 7: Accuracy is calculated.

Step 8: Comparison is done between different models.

CHAPTER - 6

MODELS USED

6.1 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 models, we used the Doc2Vec embeddings described above.

The Naive Bayes Rule is based on the Bayes' theorem

$P(c|x) = P(x|c)P(c)/ P(x)$

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.

6.2 Logistic Regression

Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. It measures the relationship between the dependent variable (TenyearCHD) and the one or more independent variables (risk factors) by estimating probabilities using underlying logistic function (sigmoid function). Sigmoid function is used as a cost function to limit the hypothesis of logistic regression between 0 and 1 (squashing) i.e. $0 \leq h\theta (x) \leq 1$.

In logistic regression cost function is defined as:

$Cost(\text{h}\theta(\text{x}),\text{y}) = \{-\log(h\theta(x)) \quad if\ y = 1\ -\log(1 - h\theta(x))\ if\ y = 0$

Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination techniques.

6.3 Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

6.4 Decision Tree

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

6.5 Support Vector Machine

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. The main idea of the SVM is to separate different classes of data by the widest "street". Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well

.

CHAPTER-7

EXPERIMENT ANALYSIS

7.1 Sample Input

The dataset contains 4 columns

1. Title
2. Text
3. Subject
4. Date

True.csv

title	text	subject	date
As U.S. budget fight looms, Republicans flip their fiscal script	WASHINGTON (Reuters) - The head of a conservative Repu	politicsNew	31-Dec-17
U.S. military to accept transgender recruits on Monday: Pentagon	WASHINGTON (Reuters) - Transgender people will be allow	politicsNew	29-Dec-17
Senior U.S. Republican senator: 'Let Mr. Mueller do his job'	WASHINGTON (Reuters) - The special counsel investigation	politicsNew	31-Dec-17
FBI Russia probe helped by Australian diplomat tip-off: NYT	WASHINGTON (Reuters) - Trump campaign adviser George	politicsNew	30-Dec-17
Trump wants Postal Service to charge 'much more' for Amazon shipmen	SEATTLE/WASHINGTON (Reuters) - President Donald Trum	politicsNew	29-Dec-17
White House, Congress prepare for talks on spending, immigration	WEST PALM BEACH, Fla./WASHINGTON (Reuters) - The Wh	politicsNew	29-Dec-17
Trump says Russia probe will be fair, but timeline unclear: NYT	WEST PALM BEACH, Fla (Reuters) - President Donald Trump	politicsNew	29-Dec-17
Factbox: Trump on Twitter (Dec 29) - Approval rating, Amazon	The following statements were posted to the verified Twitt	politicsNew	29-Dec-17
Trump on Twitter (Dec 28) - Global Warming	The following statements were posted to the verified Twitt	politicsNew	29-Dec-17
Alabama official to certify Senator-elect Jones today despite challenge:	WASHINGTON (Reuters) - Alabama Secretary of State John	politicsNew	28-Dec-17
Jones certified U.S. Senate winner despite Moore challenge	(Reuters) - Alabama officials on Thursday certified Democr	politicsNew	28-Dec-17
New York governor questions the constitutionality of federal tax overha	NEW YORK/WASHINGTON (Reuters) - The new U.S. tax coc	politicsNew	28-Dec-17
Factbox: Trump on Twitter (Dec 28) - Vanity Fair, Hillary Clinton	The following statements were posted to the verified Twitt	politicsNew	28-Dec-17
Trump on Twitter (Dec 27) - Trump, Iraq, Syria	The following statements were posted to the verified Twitt	politicsNew	28-Dec-17

Fake.csv

itle	text	subject	date
Donald Trump Sends Out Embarrassing New Year's Eve Messa	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had t	News	31-Dec-17
Drunk Bragging Trump Staffer Started Russian Collusion Inves	House Intelligence Committee Chairman Devin Nunes is going to have a bad day. He s been under the a	News	31-Dec-17
Sheriff David Clarke Becomes An Internet Joke For Threatenir	On Friday, it was revealed that former Milwaukee Sheriff David Clarke, who was being considered for I	News	30-Dec-17
Trump Is So Obsessed He Even Has Obama's Name Coded Int	On Christmas day, Donald Trump announced that he would be back to work the following day, but he	News	29-Dec-17
Pope Francis Just Called Out Donald Trump During His Christr	Pope Francis used his annual Christmas Day message to rebuke Donald Trump without even mentionin	News	25-Dec-17
Racist Alabama Cops Brutalize Black Boy While He Is In Handc	The number of cases of cops brutalizing and killing people of color seems to see no end. Now, we have	News	25-Dec-17
Fresh Off The Golf Course, Trump Lashes Out At FBI Deputy D	Donald Trump spent a good portion of his day at his golf club, marking the 84th day he s done so since	News	23-Dec-17
Trump Said Some INSANELY Racist Stuff Inside The Oval Offic	In the wake of yet another court decision that derailed Donald Trump s plan to bar Muslims from ente	News	23-Dec-17
Former CIA Director Slams Trump Over UN Bullying, Openly Si	Many people have raised the alarm regarding the fact that Donald Trump is dangerously close to beco	News	22-Dec-17
WATCH: Brand-New Pro-Trump Ad Features So Much A** Kis	Just when you might have thought we d get a break from watching people kiss Donald Trump s ass and	News	21-Dec-17
Papa John's Founder Retires, Figures Out Racism Is Bad For Bu	A centerpiece of Donald Trump s campaign, and now his presidency, has been his white supremacist w	News	21-Dec-17
WATCH: Paul Ryan Just Told Us He Doesn't Care About Strugg	Republicans are working overtime trying to sell their scam of a tax bill to the public as something that	News	21-Dec-17
Bad News For Trump – Mitch McConnell Says No To Repeali	Republicans have had seven years to come up with a viable replacement for Obamacare but they faile	News	21-Dec-17

7.2 Sample Code

7.2.1 Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
```

```
from sklearn import feature_extraction, linear_model,
model_selection, preprocessing
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
```

7.2.2 Read datasets

```
fake = pd.read_csv("Fake.csv")
true = pd.read_csv("True.csv")
```

7.2.3 Data cleaning and preparation

```
fake['target'] = 'fake'
true['target'] = 'true'
data = pd.concat([fake, true]).reset_index(drop = True)
data.shape
from sklearn.utils import shuffle
data = shuffle(data)
data = data.reset_index(drop=True)
data.drop(["date"],axis=1,inplace=True)
data.head()
data.drop(["title"],axis=1,inplace=True)
data.head()
data['text'] = data['text'].apply(lambda x: x.lower())
data.head()
import string

def punctuation_removal(text):
    all_list = [char for char in text if char not in string.punctuation]
```

```
    clean_str = ''.join(all_list)
    return clean_str

data['text'] = data['text'].apply(punctuation_removal)
data.head()

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')

data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
data.head()
```

7.2.4 Basic data exploration

```
# How many articles per subject?
print(data.groupby(['subject'])['text'].count())
data.groupby(['subject'])['text'].count().plot(kind="bar")
plt.show()

# How many fake and real articles?
print(data.groupby(['target'])['text'].count())
data.groupby(['target'])['text'].count().plot(kind="bar")
plt.show()

# Most frequent words counter
from nltk import tokenize

token_space = tokenize.WhitespaceTokenizer()
```

```
def counter(text, column_text, quantity):
    all_words = ' '.join([text for text in text[column_text]])
    token_phrase = token_space.tokenize(all_words)
    frequency = nltk.FreqDist(token_phrase)
    df_frequency = pd.DataFrame({"Word": list(frequency.keys()),
                                 "Frequency": list(frequency.values())})
    df_frequency = df_frequency.nlargest(columns = "Frequency", n = quantity)
    plt.figure(figsize=(12,8))
    ax = sns.barplot(data = df_frequency, x = "Word", y = "Frequency", color = 'blue')
    ax.set(ylabel = "Count")
    plt.xticks(rotation='vertical')
    plt.show()

   # Most frequent words in fake news
counter(data[data["target"] == "fake"], "text", 20)

# Most frequent words in real news
counter(data[data["target"] == "true"], "text", 20)
```

7.2.5 Modeling

```
# Function to plot the confusion matrix
import itertools

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
```

```
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

7.2.6 Preparing the data

```
# Split the data
X_train,X_test,y_train,y_test = train_test_split(data['text'], data.target, test_size=0.2, random_state=42)
```

7.2.7 Models

We will implement five models here and compare their performance.

7.2.7.1 Naive Bayes

```
dct = dict()
```

```
from sklearn.naive_bayes import MultinomialNB

NB_classifier = MultinomialNB()
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', NB_classifier)])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))

dct['Naive Bayes'] = round(accuracy_score(y_test, prediction)*100,2)
```

7.2.7.2 Logistic Regression

```
# Vectorizing and applying TF-IDF
from sklearn.linear_model import LogisticRegression

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])

# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Logistic Regression'] = round(accuracy_score(y_test, prediction)*100,2)
```

7.2.7.3 Random Forest

```
from sklearn.ensemble import RandomForestClassifier

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Random Forest'] = round(accuracy_score(y_test, prediction)*100,2)
```

7.2.7.3 Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

# Vectorizing and applying TF-IDF
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier(criterion= 'entropy',
                                                  max_depth = 20,
                                                  splitter='best',
                                                  random_state=42))])
# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Decision Tree'] = round(accuracy_score(y_test, prediction)*100,2)
```

7.2.7.4 SVM

```
from sklearn import svm
```

```
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', clf)])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['SVM'] = round(accuracy_score(y_test, prediction)*100,2)
```

7.2.8 Comparing different models

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,7))
plt.bar(list(dct.keys()),list(dct.values()))
plt.ylim(90,100)
plt.yticks((91, 92, 93, 94, 95, 96, 97, 98, 99, 100))
```

7.3 Performance

Model Name	Accuracy
Naïve Bayes	95.80%
Logistic Regression	99.10%
Decision Tree	99.57%
Random Forest	99.33%
Support Vector Machine	99.62%

As we can see, the decision tree classifier performed the best on the train set and gave an accuracy of 99.57%

CHAPTER – 8

TESTING

The lack of manually labeled fake news datasets is certainly a bottleneck for advancing computationally intensive, text-based models that cover a wide array of topics. The dataset for the fake news challenge does not suit our purpose due to the fact that it contains the ground truth regarding the relationships between texts but not whether or not those texts are actually true or false statements. For our purpose, we need a set of news articles that is directly classified into categories of news types (i.e. real vs. fake or real vs parody vs. clickbait vs. propaganda). For more simple and common NLP classification tasks, such as sentiment analysis, there is an abundance of labeled data from a variety of sources including Twitter, Amazon Reviews, and IMDb Reviews. Unfortunately, the same is not true for finding labeled articles of fake and real news. This presents a challenge to researchers and data scientists who want to explore the topic by implementing supervised machine learning techniques. I have researched the available datasets for sentence-level classification and ways to combine datasets to create full sets with positive and negative examples for document-level classification**.**

- INSTALLATION TESTING

There exists no dataset of similar quality to the Liar Dataset for document level classification of fake news. As such, I had the option of using the headlines of documents as statements or creating a hybrid dataset of labeled fake and legitimate news articles. This shows an informal and exploratory analysis carried out by combining two datasets that individually contain positive and negative fake news examples. Genes trains a model on a specific subset of both the Kaggle dataset and

the data from NYT and the Guardian. In his experiment, the topics involved in training and testing are restricted to U.S News, Politics, Business and World news. However, he does not account for the difference in date range between the two datasets, which likely adds an additional layer of topic bias based on topics that are more or less popular during specific periods of time. We have collected data in a manner similar to that of Genes , but more cautious in that we control for more bias in the sources and topics. Because the goal of our project was to find patterns in the language that are indicative of real or fake news, having source bias would be detrimental to our purpose. Including any source bias in our dataset, i.e. patterns that are specific to NYT, The Guardian, or any of the fake news websites, would allow the model to learn to associate sources with real/fake news labels. Learning to classify sources as fake or real news is an easy problem, but learning to classify specific types of language and language patterns as fake or real news is not. As such, we were very careful to remove as much of 15 the sourcespecific patterns as possible to force our model to learn something more meaningful and generalizable. We admit that there are certainly instances of fake news in the New York Times and probably instances of real news in the Kaggle dataset because it is based on a list of unreliable websites. However, because these instances are the exception and not the rule, we expect that the model will learn from the majority of articles that are consistent with the label of the source. Additionally, we are not trying to train a model to learn facts but rather learn deliveries. To be more clear, the deliveries and reporting mechanisms found in fake news articles within New York Times should still possess characteristics more commonly found in real news, although they will contain fictitious factual information.

- COMPATIBILITY TESTING

The system uses a dataset of fake news articles that was gathered by using a tool called th BS detector which essentially has a blacklist of websites that are sources of fake news. The articles were all published in the 30 days, While any span of dates would be characterized by the current events of that time, this range of dates is particularly interesting because it spans the time directly before, during, and directly after the 2016 election. The dataset has articles and metadata from 244 different websites, which is helpful in the sense that the variety of sources will help the model to not learn a source bias. However, at a first glance of the dataset, you can easily tell that there are still certain obvious reasons that a model could learn specifics of what is included in the “body” text in this dataset. For example, there are instances of the author and source in the body text , Also, there are some patterns like including the date that, if not also repeated in the real news dataset, could be learned by the model.

CHAPTER -9

CONCLUSION

The task of classifying news manually requires in-depth knowledge of the domain and expertise to identify anomalies in the text. In this research, we discussed the problem of classifying fake news articles using machine learning models and ensemble techniques. The data we used in our work is collected from the World Wide Web and contains news articles from various domains to cover most of the news rather than specifically classifying political news. The primary aim of the research is to identify patterns in text that differentiate fake articles from true news. We extracted different textual features from the articles using an LIWC tool and used the feature set as an input to the models. The learning models were trained and parameter-tuned to obtain optimal accuracy. Some models have achieved comparatively higher accuracy than others. We used multiple performance metrics to compare the results for each algorithm. The ensemble learners have shown an overall better score on all performance metrics as compared to the individual learners.

Fake news detection has many open issues that require attention of researchers. For instance, in order to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Graph theory and machine learning techniques can be employed to identify the key sources involved in spread of fake news. Likewise, real time fake news identification in videos can be another possible future direction.

REFERENCES

1. machinelearningmastery.com

2. python.org

3. FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network by Jiawei Zhang1 , Bowen Dong2 , Philip S. Yu2

4. Zhang, X. and A.A. Ghorbani, An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, 2019.

5.] Zhang, C., et al., Detecting fake news for reducing misinformation risks using analytics approaches. European Journal of Operational Research, 2019.

6. Aldwairi, M. and A. Alwahedi, Detecting Fake News in Social Media Networks. Procedia Computer Science, 2018. 141: p. 215-222

7. Fake News Detection on Social Media-A Review by Steni Mol TS and Sreeja PS

8. Fake News Detection on Social Media: A Data Mining Perspective Kai Shu , Amy Sliva, Suhang Wang , Jiliang Tang , and Huan Liu

9. Dataset from Keggle

10. Fake News Detection: A Data Mining Perspective