

# Unveiling Political Power in Textual Parliamentary Debate using Machine Learning and Deep Learning Techniques

submitted by

Suchanda Datta

Class Roll No: 002210503033

Exam Roll: MCA244032

Master of Computer Application

under the supervision of

Dr. Dipankar Das

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

2024

Faculty of Engineering and Technology  
Jadavpur University

**CERTIFICATE OF RECOMMENDATION**

**To Whom It May Concern**

I hereby recommend that the project report entitled “Unveiling Political Power in Textual Parliamentary Debate using Machine Learning and Deep Learning Techniques” has been carried out by Suchanda Datta (Reg. No.: 163625 of 2022-23, Roll No.: 002210503033, Exam Roll: MCA244032), under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic year 2022-2024.

---

Dr. Dipankar Das  
Project Supervisor  
Dept. of Computer Science and  
Engineering  
Jadavpur University

---

Prof. (Dr.) Nandini Mukhopadhyay  
Head of the Department  
Dept. of Computer Science & Engineering  
Jadavpur University

---

Prof. Dipak Laha  
Dean, Faculty Council  
of Engineering & Technology  
Jadavpur University, Kolkata-032

Faculty of Engineering and Technology  
Jadavpur University

**CERTIFICATE OF APPROVAL**

This is to certify that the project report entitled **“Unveiling Political Power in Textual Parliamentary Debate using Machine Learning and Deep Learning Techniques”** is a bonafide record of work carried out by SUCHANDA DATTA, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

---

Signature of Examiner 1

Date :

---

Signature of Examiner 2

Date :

Faculty of Engineering and Technology  
Jadavpur University

**DECLARATION OF COMPLIANCE WITH  
ACADEMIC ETHICS**

I hereby declare that this project report entitled "Unveiling Political Power in Textual Parliamentary Debate using Machine Learning and Deep Learning Techniques" is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, all materials and results that are not original to this work have been fully cited and referenced.

Suchanda Datta

Registration No.: 163625 of 2022-23

Examination Roll No.: MCA244032

Thesis Title :- Unveiling Political Power in Textual Parliamentary Debate using Machine Learning and Deep Learning Techniques

---

Signature with Date

# **ACKNOWLEDGEMENT**

With my most sincere and gratitude, I would like to thank Dr. Dipankar Das, Department of Computer Science & Engineering, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestions and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providing sound knowledge base and cooperation.

I would like to thank all the faculty members of the Department of Computer Science & Engineering of Jadavpur University for their continuous support.

Last, but not the least, I would like to thank my batch mates for staying by my side when I needed them.

---

Suchanda Datta

Class Roll No.: 002210503033

Examination Roll No.: MCA244032

Department of Computer Science & Engineering

Jadavpur University

# Contents

|                                                        |    |
|--------------------------------------------------------|----|
| List of figures.....                                   | 2  |
| List of tables .....                                   | 3  |
| Abstract .....                                         | 4  |
| Chapter 1 : Introduction .....                         | 5  |
| 1.1 Overview .....                                     | 5  |
| 1.1.1 Five phases of Natural Language Processing ..... | 5  |
| 1.1.2 Sentiment Analysis.....                          | 6  |
| 1.2 Background Review and Motivation.....              | 7  |
| 1.3 Objectives .....                                   | 8  |
| 1.4 Contributions .....                                | 10 |
| 1.5 Outline of the Report.....                         | 10 |
| Chapter 2 : Literary Survey.....                       | 12 |
| Chapter 3 : Methodology .....                          | 13 |
| 3.1 Problem Statement .....                            | 13 |
| 3.2 Proposed Technique/ Approach .....                 | 13 |
| Chapter 4 : Implementation and Results .....           | 24 |
| 4.1 Datasets .....                                     | 24 |
| 4.2 Experiments and Results.....                       | 26 |
| 4.3 Error Analysis .....                               | 28 |
| Chapter 5 : Conclusion and Future Work .....           | 33 |
| References .....                                       | 34 |

## List of figures

|                                                                                                                                                                                                                                       |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1 : Word Cloud Generated from Danish Parliamentary Debate Dataset .....                                                                                                                                                        | 15 |
| Figure 2 : Word Cloud Generated from Indian Parliamentary Debate Dataset .....                                                                                                                                                        | 15 |
| Figure 3 : Schematic Diagram of Logistic Regression .....                                                                                                                                                                             | 17 |
| Figure 4 : Schematic Diagram of Support Vector Machine .....                                                                                                                                                                          | 18 |
| Figure 5 : Schematic Diagram of Gaussian Naive Bayes Classifier .....                                                                                                                                                                 | 18 |
| Figure 6 : Schematic Representation of K-Nearest Neighbor (KNN) algorithm concept. Where, $k$ = number of neighbours; $x_1$ & $x_2$ are the input descriptors; red star is the query instance..                                       | 19 |
| Figure 7 : Schematic Diagram of Random Forest Classifier .....                                                                                                                                                                        | 20 |
| Figure 8 : Recurrent Neural Network .....                                                                                                                                                                                             | 21 |
| Figure 9 : Gated Neural Network .....                                                                                                                                                                                                 | 21 |
| Figure 10 : Long Short Term Memory .....                                                                                                                                                                                              | 22 |
| Figure 11 : Describes the architecture of the BI-LSTM layer where $x_i$ is the input token, $y_i$ is the output token, and $a$ and $a'$ are lstm nodes. The final output of $y_i$ is the combination of $a$ and $a'$ LSTM nodes. .... | 23 |
| Figure 12 : Elbow Method used in Danish Parliamentary Debate Dataset Classification .....                                                                                                                                             | 29 |
| Figure 13 : Elbow Method used in Indian Parliamentary Debate Dataset Classification .....                                                                                                                                             | 30 |

## List of tables

|                                                                                                                   |       |
|-------------------------------------------------------------------------------------------------------------------|-------|
| Table 1 : Result of Machine Learning Models .....                                                                 | 26    |
| Table 2 : Result of Machine Learning Models on Indian Dataset .....                                               | 27    |
| Table 3 : Result of Deep Learning Models on Danish Dataset .....                                                  | 27    |
| Table 4 : Confusion Matrices from Danish Parliamentary Debate Dataset Classification using Machine Learning ..... | 28    |
| Table 5 : Confusion Matrices from Indian Parliamentary Debate Dataset Classification using Machine Learning ..... | 29-30 |
| Table 6 : Confusion Matrices from Danish Parliamentary Debate Dataset Classification using Deep Learning .....    | 31-32 |

# Abstract

---

A parliamentary debate can be described in general terms as a formal discussion on a particular topic that is strictly controlled by an institutional set of rules and presided over by the Speaker of the House. Debates in national parliaments do not only affect the fundamental aspects of citizens' life, but often a broader area, or even the whole world. As a form of political debate, however, parliamentary speeches are often indirect and present a number of challenges to computational analyses. In this task, we focus on identifying whether the speaker in the parliamentary debate belong to a governing party or a party in opposition. This task is formulated as a binary classification tasks.

This report comprises of two parts. In the first part, we have applied Sentiment Analysis on a labelled dataset of Danish parliamentary debate, by using five Machine learning and four Deep learning models, and have done a comparative study of the performance of each of these models. For Danish parliamentary debate dataset, the highest score that we got among the five Machine Learning models is that of Support Vector Machine (accuracy = 74%) and lowest score is of Gaussian Naïve Bayes (accuracy = 53%), and the highest score that we got among the four Deep Learning models is that of Bi-LSTM (accuracy = 67%) and lowest score is of LSTM (accuracy = 39%).

Although European parliamentary debate datasets are available on the Internet, there are no such dataset available on Indian parliamentary debate that's why we have decided to prepare one. In the second part, we have prepared a dataset of Indian parliamentary debate from the unstructured pdf files of Lok Sabha debate, available in the Parliament Digital Library website, keeping all the fields of this dataset similar to that of Danish parliamentary debate dataset and then we have applied Sentiment Analysis on our prepared dataset by using five Machine learning models, and have done a comparative study of the performance of each of these models.

But unlike Danish parliamentary debate dataset, here we have to identify whether the speaker in the parliamentary debate belong to a governing party or a party in opposition or neither of them. This task is formulated as a three class classification tasks. For Indian parliamentary debate dataset, the highest score that we got among the five Machine Learning models is that of K-Nearest Neighbour (accuracy = 64%) and lowest score is of Gaussian Naïve Bayes and K-Nearest Neighbour with K=1 (accuracy = 41%).

# Chapter 1 : Introduction

---

## 1.1 Overview

Natural Language Processing (NLP) is a field that combines computer science, linguistics, and machine learning to study how computers and humans communicate in natural language. The goal of NLP is for computers to interpret and generate human language, improving work efficiency and facilitating interaction with machines. NLP bridges the gap between humans and electronic devices.

As a subfield of artificial intelligence, NLP uses computational techniques to process and analyze natural language data, such as text and speech, aiming to understand the meaning behind the language. It is used in various applications, including machine translation, sentiment analysis, speech recognition, chatbots, and text classification. By leveraging computational linguistics, statistics, machine learning, and deep learning, NLP enables computers to comprehend human language, including the intentions and emotions of speakers or writers.

### 1.1.1 Five phases of Natural Language Processing

#### 1) Lexical Analysis

A lexicon is the vocabulary that forms a language. Lexical analysis breaks down language into units, or lexemes, such as paragraphs, sentences, phrases, and words. NLP algorithms categorize words into parts of speech (POS) and further divide lexemes into morphemes, the smallest meaningful language units. Morphemes come in two types: Free morphemes can stand alone as words (e.g., "cow," "house"). Bound morphemes form larger words, like "unimaginable," which includes "un-" (negative prefix), "imagine" (root), and "-able" (suffix indicating ability). For example, lexical analysis of the first sentence separates it into phrases like "the understandable vocabulary that makes up a language," then into word lexemes like "vocabulary" and "language" (nouns), and finally into free morphemes like "words" and "vocabulary," and bound morphemes like "-able."

#### 2) Syntactic Analysis

Syntax refers to the arrangement of words and phrases to create well-formed sentences in a language. Syntactic analysis evaluates these arrangements to ensure grammatical correctness. For example, the sentence "Dave wrote the paper" is

grammatically correct and passes syntactic analysis. In contrast, a sentence like “Dave do jumps” is identified as syntactically incorrect during analysis.

### **3) Semantic Analysis**

Semantics pertains to the meanings of words, phrases, sentences, and paragraphs. Semantic analysis aims to understand the literal meaning of individual language elements rather than their syntactic correctness. However, it does not consider the surrounding context to clarify the meaning. For example, the sentence "Manhattan calls out to Dave" is grammatically correct and passes syntactic analysis. However, it fails semantic analysis because Manhattan, being a place, cannot literally call out to someone, rendering the sentence nonsensical in meaning.

### **4) Discourse Integration**

Discourse refers to communication between two or more individuals. Discourse integration examines previous words and sentences to clarify the meaning of ambiguous language.

For example, if one sentence says, "Manhattan speaks to all its people," and the next says, "It calls out to Dave," discourse integration uses the context from the first sentence to determine that "It" in the second sentence refers to Manhattan.

### **5) Pragmatic Analysis**

Pragmatism involves interpreting the intended meaning of language. Pragmatic analysis seeks to determine the intended, rather than the literal, meaning of words and phrases.

For example, a pragmatic analysis can reveal the intended meaning behind the sentence "Manhattan speaks to all its people." Techniques such as neural networks analyze the context to recognize that the sentence is not meant to be taken literally. Instead, they interpret it as a metaphor for the emotional connection people have with places.

## **1.1.2 Sentiment Analysis**

Sentiment analysis, also called opinion mining, is the field of study that analyzes people’s opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text. The entities can be products, services, organizations, individuals, events, issues, or topics. The field represents a large problem space. Many related names and slightly different tasks – for example, sentiment analysis, opinion mining, opinion analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, and review mining – are now all under the umbrella of sentiment analysis.

The primary objective of sentiment analysis is to comprehend the sentiment enclosed within a text, whether positive, negative, or neutral. Sentiment analysis is a process that involves analyzing textual data such as social media posts, product reviews, customer feedback, news articles, or any other form of text to classify the sentiment expressed in the text. The sentiment can be classified into three categories: Positive Sentiment Expressions indicate a favorable opinion or satisfaction; Negative Sentiment Expressions indicate dissatisfaction, criticism, or negative views; and Neutral Sentiment Text expresses no particular sentiment or is unclear.

Before analyzing the text, some preprocessing steps usually need to be performed. These include tokenization, breaking the text into smaller units like words or phrases, removing stop words such as common words like “and,” “the,” and so on, and stemming or lemmatization, which involves reducing words to their base or root form. At a minimum, the data must be cleaned to ensure the tokens are usable and trustworthy.

## **1.2 Background Review and Motivation**

In the digital era, online platforms have become significant arenas for parliamentary debates, supplementing traditional offline discussions. Understanding power dynamics within these debates is crucial for deciphering influence dissemination and policy formulation. Parliamentary debates, once confined to legislative chambers, now extend to online forums and social networks, broadening participation and amplifying discourse.

Social networks play a pivotal role in shaping public discourse, influencing opinions, and facilitating diverse viewpoints, offering rich data for sentiment analysis and power identification. Power identification within parliamentary debates entails discerning the influential actors, persuasive arguments, and dominant narratives that shape the discourse landscape. By analyzing the sentiment and linguistic patterns inherent in parliamentary speeches, comments, and interactions, researchers can uncover latent power structures and hierarchies within debate forums.

Understanding these power dynamics is crucial for elucidating the mechanisms through which influence is exerted, opinions are swayed, and decisions are made. This research aims to leverage sentiment analysis, Natural Language Processing, and computational techniques to: Illuminate power structures, Uncover sentiment trends and Enhance decision-making processes.

Power identification in parliamentary debates involves discerning influential actors, persuasive narratives, and dominant dynamics within the discourse. This can be achieved through various means, such as analyzing speaking patterns to identify influential speakers and assessing the frequency and impact of contributions from different political parties. Additionally, studying how speakers respond to each other's arguments, mapping social

network connections between parliamentary members, and tracking media coverage and public reactions can unveil hidden power structures and alliances.

By leveraging sentiment analysis and observing legislative outcomes resulting from debates, researchers can attribute credit or responsibility for successful policy initiatives or amendments. Overall, power identification techniques offer valuable insights into the distribution and exercise of influence within parliamentary debates, aiding analyses of political dynamics, decision-making processes, and policy outcomes.

As mentioned in the website – “Touche at CLEF 2024 – Webis Group”, Debates in national parliaments have far-reaching implications, influencing not only the fundamental aspects of citizens' lives but also impacting broader regions and, at times, the entire world. Parliamentary speeches, as a form of political discourse, are critical to understanding legislative priorities and policy directions. However, these speeches often exhibit a high degree of indirectness and complexity, presenting substantial challenges for computational analysis.

The analysis of parliamentary speeches holds significant value in political science and public policy. By understanding the nuances of these debates, researchers can gain insights into the ideological stances of political actors, track changes in policy preferences over time, and assess the dynamics between governing and opposition parties. Moreover, in an era where large volumes of text data are readily available, computational methods offer a scalable approach to dissecting these intricate speeches.

Our task is motivated by the need to develop robust computational models that can accurately identify a crucial variable associated with speakers in a parliamentary debate: that whether they are affiliated with the governing party or the opposition. Accurate identification of this variable can enhance our understanding of parliamentary dynamics, inform public discourse, and contribute to more transparent and accountable governance.

## 1.3 Objectives

The objectives of this report are divided into two parts.

- **The first part is to build some models that can classify the speeches of speakers in the dataset of Danish parliamentary debate into two classes, namely class 1 which specifies that the speech belongs to the speaker of ruling party and class 0 which specifies that the speech belongs to the speaker of opposition party.**

An example of Class 1 (Ruling Party Speaker) :

**Speech:**

"Today, we take a significant step forward in improving the nation's healthcare system. Our government is committed to ensuring that every citizen has access to

affordable and high-quality medical services. We have allocated additional funding to public hospitals and introduced new policies to reduce waiting times and improve patient care. This reform will bring about a positive change and strengthen the welfare state that we all cherish."

**Label** : Class 1 (Ruling Party Speaker)

An example of Class 0 (Opposition Party Speaker) :

**Speech:**

"The current government's healthcare reform is a short-sighted approach that fails to address the root problems in our medical system. By merely increasing funding without tackling inefficiencies and corruption, they are wasting taxpayer money. We need a comprehensive plan that includes accountability measures and a focus on preventive care. The opposition stands firm in advocating for a more sustainable and transparent healthcare strategy that truly benefits the people."

**Label** : Class 0 (Opposition Party Speaker)

- **The second part is to prepare a dataset for Indian parliamentary debate, similar to Danish one. And then build some models that can classify the speeches of speaker in the dataset of Indian parliamentary debate into class 0 (opposition party speaker) , class 1 (ruling party speaker) and class 2 (neither ruling nor opposition party member).**

An example of Class 1 (Ruling Party Speaker) :

**Speech:**

"Our government is dedicated to advancing the nation's infrastructure. We have launched several projects aimed at modernizing our roads, railways, and ports. These developments will not only boost our economy but also create millions of jobs for our citizens. We are committed to building a stronger, more prosperous India."

**Label** : Class 1 (Ruling Party Speaker)

An example of Class 0 (Opposition Party Speaker) :

**Speech:**

"The current administration's infrastructure projects are marred by delays and cost overruns. While the government boasts about creating jobs, the reality is that many projects are stalled and workers are left in uncertainty. We need transparency and better planning to ensure that these initiatives truly benefit the people."

**Label** : Class 0 (Opposition Party Speaker)

An example of Class 2 (Neither Ruling Nor Opposition Party Member) :

**Speech:**

"As an independent member, I believe it's crucial to focus on sustainable development. While infrastructure is important, we must also consider the environmental impact of our projects. I urge the parliament to adopt greener practices and invest in renewable energy sources to ensure a balanced and sustainable growth."

**Label** : Class 2 (Neither Ruling Nor Opposition Party Member)

## 1.4 Contributions

In order to achieve the first objective, we divided the labelled dataset of Danish Parliamentary debate into training set (80%) and testing set (20%), and then train five Machine Learning (Logistic Regression<sup>1</sup>, Support Vector Machine<sup>2</sup>, Gaussian Naïve Bayes<sup>3</sup>, K-Nearest Neighbour<sup>4</sup>, Random Forest<sup>5</sup>) and four Deep Learning models (Simple Recurrent Neural Network<sup>6</sup>, Gated Recurrent Unit, Long Short Term Memory<sup>7</sup>, Bi-LSTM<sup>8</sup>) on that training set, and after that we test the models on testing set.

Finally, we figure out the model/models that achieves highest accuracy. To achieve our second objective, we followed some pre-processing steps to convert the pdf file of Lok Sabha debate, which is in unstructured format into a structured pdf file containing the data in a particular format that is suitable for it's retrieval (like speaker name : speaker's speech). Then we designed a python script<sup>9</sup> that takes this formatted pdf file as input and gives the final dataset in excel sheet as output. We try out all the Machine Learning models that we previously used, on this dataset of Indian parliamentary debate, and just as we did previously, we figure out the model/models with highest accuracy.

## 1.5 Outline of the Report

In Chapter 2, we have the Literature survey on five research papers.

In Chapter 3, we have written the Methodology, which includes specifying the problem statement and the proposed technique and approach.

---

<sup>1</sup><https://gist.github.com/pb111/62067d128be7f7f86e81916ff946af7b>

<sup>2</sup> [https://github.com/mahesh147/Support-Vector-Machine/blob/master/support\\_vector\\_machine.py](https://github.com/mahesh147/Support-Vector-Machine/blob/master/support_vector_machine.py)

<sup>3</sup> <https://github.com/bamtak/machine-learning-implementatation-python/blob/master/Gaussian%20Naive%20Bayes.ipynb>

<sup>4</sup> <https://github.com/vivek2319/K-Nearest-Neighbors/blob/master/Knn.py>

<sup>5</sup> <https://gist.github.com/pb111/88545fa33780928694388779af23bf58#file-random-forest-classification-with-python-and-scikit-learn-ipynb>

<sup>6</sup> <https://github.com/Vnicius/simple-rnn/blob/master/rnn.py>

<sup>7</sup> [https://github.com/omerbsezer/LSTM\\_RNN\\_Tutorials\\_with\\_Demo/blob/master/BasicLSTM/main.py](https://github.com/omerbsezer/LSTM_RNN_Tutorials_with_Demo/blob/master/BasicLSTM/main.py)

<sup>8</sup> <https://github.com/orlandxrf/spanish-ner/blob/master/bilstm.py>

<sup>9</sup> <https://colab.research.google.com/drive/1IOYiLrTSC5yF4wvCpBJbP57hCX8hLKG?usp=sharing>

In Chapter 4, we have written the implementation and result part, here we have given a brief description of our datasets. In this chapter, we have also shown our experimental results and have done the error analysis.

In Chapter 5, we have discussed the conclusion and future scope of our project report.

## Chapter 2 : Literary Survey

---

The exploration of sentiment analysis techniques and visualization mechanisms within the realm of political debates has garnered considerable attention from researchers seeking to unveil underlying power dynamics and decision-making processes. The following literature review provides insights into relevant studies and methodologies, culminating in an understanding of the landscape in which the present research, denoted as [1], operates.

To begin, a foundational aspect of sentiment analysis lies in the classification-based and lexicon-based approaches. In [1], the author delves into these methodologies, assessing their efficacy in predicting debate outcomes and facilitating visualization. This examination aligns with prior works in sentiment analysis, where researchers have explored various classification algorithms and lexicon-based methods to analyze textual data from diverse domains. Notably, studies such as [2] and [3] have investigated the application of machine learning algorithms, including support vector machines and deep learning models, in sentiment classification tasks across different contexts, laying the groundwork for comparative analysis [1].

Political debates often trigger national controversies, shaping public sentiment and fostering polarization, especially with the proliferation of social media platforms like Twitter. This trend enables individuals to express diverse viewpoints, facilitating the analysis of communicative behaviors and interactions among users with varied societal roles and political sentiments. A notable study focused on Twitter communications during the 2012-2013 debate over the marriage reform in France, known as "Le Mariage Pour Tous," exemplifies this endeavor [4]. Through linguistic and non-linguistic analyses of tweets tagged with #mariagepourtous, the study examined subjective language usage and the impact of diverse user types, shedding light on the dynamics of virtual debates and stakeholder influence in shaping public opinion.

Current sentiment analysis tools for social text messages, tweets, blogs, etc., are often slow and computationally heavy. To address these inefficiencies, a novel method using Partial Textual Entailment (PTE) is proposed by [5]. PTE measures semantic similarity between tweets to group similar ones together, reducing the burden on sentiment analyzers and speeding up processing. Additionally, a modification to PTE enhances its applicability in various Natural Language Processing tasks, improving accuracy and efficiency. This method offers a more efficient and scalable approach to sentiment analysis, potentially transforming the field.

## Chapter 3 : Methodology

---

### 3.1 Problem Statement

The problem statement is divided into two sub-problems :

**Problem statement 1** : Identifying whether the speaker in the Danish parliamentary debate belong to a governing party or a party in opposition.

**Problem statement 2** :\_Preparing a dataset on Indian parliamentary debate and then identifying whether the speaker in that dataset belong to a governing party or a party in opposition.

### 3.2 Proposed Technique/ Approach

The proposed technique/approach is given below :

#### 3.2.1 Text Pre-processing Techniques

**Lowercasing** - Python is a case sensitive programming language. Therefore, to avoid any issues and ensure consistency in the processing of the text, we convert all the text to lowercase.

Example:

Original: "The Quick Brown Fox Jumps Over the Lazy Dog."

Lowercased: "the quick brown fox jumps over the lazy dog."

**Digits removal** - It is important to remove all numerical digits from the text dataset. This is because, in most cases, numerical values do not provide any significant meaning to the text analysis process.

Example:

Original: "In 2024, the population of the city is expected to reach 1.5 million."

Digits removed: "In , the population of the city is expected to reach million."

**HTML tags removal** - When building a model, HTML tags are typically not relevant and can be removed from the text data.

Example:

Original: "<p>The quick brown fox jumps over the lazy dog.</p>"

HTML tags removed: "The quick brown fox jumps over the lazy dog."

**Punctuation removal** - Removing punctuation is a foundational step in preprocessing text for Natural Language Processing (NLP) tasks. It simplifies the dataset, reducing complexity and allowing models to focus on the semantic content of the text.

Example:

Original: "Hello, world! How are you?"

Punctuation removed: "Hello world How are you"

**Additional preprocessing** – This additional preprocessing step cleans and normalizes a given text string by performing several steps:

Removal of leading and trailing whitespace, Removal of numeric citations or references, Removal of punctuation and convert to lowercase, Replacing multiple spaces with a single space. The processed text is then returned, making it consistent and easier to work with for tasks like text analysis.

Example:

Original: " The quick brown fox jumps over the lazy dog. 123 "

Processed: "the quick brown fox jumps over the lazy dog"

Removal of leading and trailing whitespace: "The quick brown fox jumps over the lazy dog. 123" → "The quick brown fox jumps over the lazy dog. 123"

Removal of numeric citations or references: No numeric citations in the example.

Removal of punctuation and convert to lowercase: "The quick brown fox jumps over the lazy dog 123" → "the quick brown fox jumps over the lazy dog"

Replacing multiple spaces with a single space: "the quick brown fox jumps over the lazy dog" → "the quick brown fox jumps over the lazy dog"

**Stop word removal** - Stopwords refer to the most commonly occurring words in any natural language. For the purpose of analyzing text data and building NLP models, these stopwords might not add much value to the meaning of the document. Therefore, removing stopwords can help us to focus on the most important information in the text and improve the accuracy of our analysis. One of the advantages of removing stopwords is that it can reduce the size of the dataset, which in turn reduces the training time required for natural language processing models.

Example:

Original: "This is an example of text preprocessing."

Stop words removed: "example text preprocessing"

**Generating word cloud from training set and then finding and removing domain specific parliamentary words –**

Following word clouds were generated from Danish parliamentary and Indian parliamentary debate dataset, after stop word removal, as shown in figure 1 and 2 respectively :



From Figure 1, we can say that the words like "government", "denmark", "minister", "danish", "parliament", "debate" are all very frequently occurring words, and similarly from Figure 2, we can say that the words like "government", "speaker sir", "law", "honorable minister", "honorable speaker", "sir", "india", "honorable", "house", "speaker", "hon", "minister", "ji", "hon minister" are all very frequently occurring words. Thus for similar reasons as for stop word removal these domain specific words were also removed.

### 3.2.2 Stemming

Stemming is the process of reducing the word to its word stem that affixes to suffixes and prefixes or to roots of words known as a lemma. In simple words stemming is reducing a word to its base word or stem in such a way that the words of similar kind lie under a common stem.

For example – The words care, cared and caring lie under the same stem 'care'. Stemming is important in natural language processing(NLP). Here we have used Snowball stemmer. Snowball Stemmer is a stemming algorithm which is also known as the Porter2 stemming algorithm as it is a better version of the Porter.

### 3.2.3 Feature extraction

**TF-IDF vectorization :** TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set).

#### **Term Frequency (TF)**

It is a measure of the frequency of a word (w) in a document (d). TF is defined as the ratio of a word's occurrence in a document to the total number of words in a document. The denominator term in the formula is to normalize since all the corpus documents are of different lengths.

The initial step is to make a vocabulary of unique words and calculate TF for each document. TF will be more for words that frequently appear in a document and less for rare words in a document.

#### **Inverse Document Frequency (IDF)**

It is the measure of the importance of a word. Term frequency (TF) does not consider the importance of words. Some words such as 'of', 'and', etc. can be most frequently present but are of little significance. IDF provides weightage to each word based on its frequency in the corpus D.

### Term Frequency — Inverse Document Frequency (TFIDF)

It is the product of TF and IDF. TFIDF gives more weightage to the word that is rare in the corpus (all the documents). TFIDF provides more importance to the word that is more frequent in the document.

## 3.2.4 Machine Learning Based Approaches

### 1) Logistic Regression

Logistic regression is an effective method for binary classification tasks like sentiment analysis. It classifies text data into categories, such as positive or negative sentiment, by converting text into numerical features using techniques like TF-IDF or word embeddings. The model is then trained on labeled data to learn the relationship between these features and sentiment labels. Once trained, the model predicts the sentiment of new text, providing probabilities for each class. Logistic regression is particularly suitable for binary classification due to its probabilistic output, which indicates prediction confidence, and its simplicity and computational efficiency, making it ideal for large datasets. It models a linear relationship between features and the log-odds of the binary outcome, supports regularization to prevent overfitting, and offers interpretable coefficients that help understand the influence of each feature. These features make logistic regression an ideal choice for sentiment analysis.

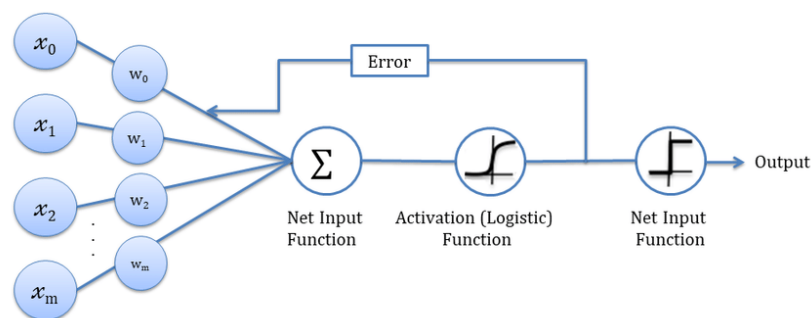


FIGURE 3 : SCHEMATIC DIAGRAM OF LOGISTIC REGRESSION

### 2) Support Vector Machine (SVM)

Support Vector Machine (SVM) is used for binary classification by finding the optimal hyperplane that separates data points of different classes, maximizing the margin between the hyperplane and the nearest data points from each class (support vectors). This approach ensures robustness and good generalization to new data. When data is not linearly separable, SVM employs kernel functions to transform the input space into a higher-dimensional space where a hyperplane can effectively separate the classes. Common kernel functions include the linear kernel for linearly

separable data, polynomial kernel for mapping to higher polynomial dimensions, radial basis function (RBF) kernel for non-linear relationships, and sigmoid kernel similar to a neural network's activation function. Given that our dataset is linearly separable, we use the linear kernel.

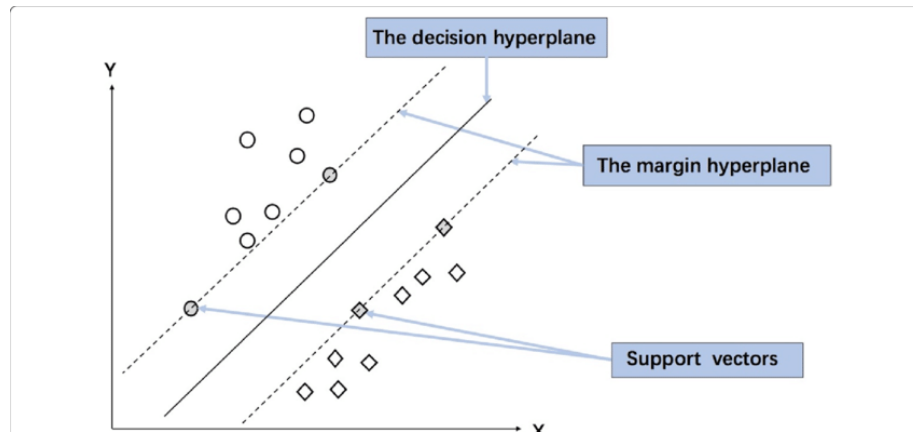


FIGURE 4 : SCHEMATIC DIAGRAM OF SUPPORT VECTOR MACHINE

### 3) Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem, which calculates the probability that a given data point belongs to a particular class using the likelihood of the features given the class, the prior probability of the class, and the evidence (overall probability of the data). It assumes that all features are independent given the class label, simplifying the computation of the likelihood, and that continuous features follow a Gaussian (normal) distribution within each class. These assumptions enable a straightforward and computationally efficient classification method that performs well in many practical applications despite its simplicity.

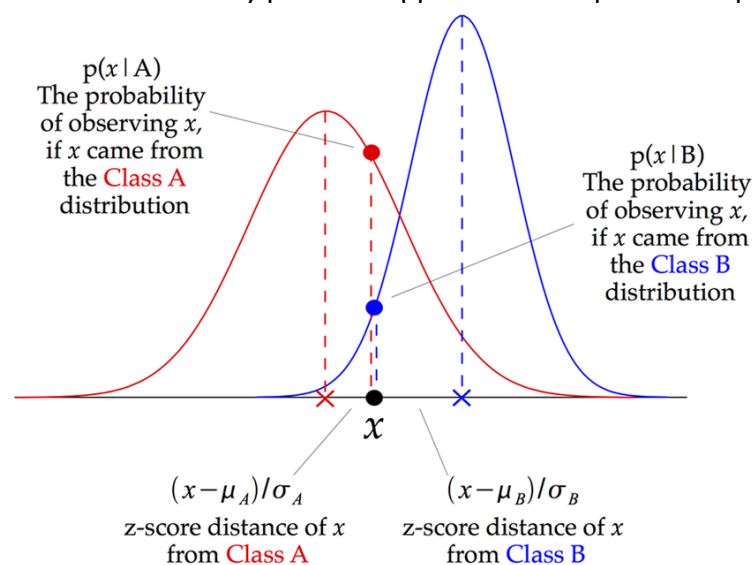


FIGURE 5 : SCHEMATIC DIAGRAM OF GAUSSIAN NAIVE BAYES CLASSIFIER

#### 4) K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) classifies a data point based on the labels of its nearest neighbors in the feature space by calculating the distance, commonly Euclidean, between data points and assigning the new data point the most common class among its  $k$  closest neighbors. The parameter  $k$ , which determines the number of neighbors considered, plays a crucial role in model performance: a small  $k$  (e.g., 1) can make the model sensitive to noise, while a large  $k$  can overly smooth class boundaries. Typically,  $k$  is chosen through cross-validation to balance bias and variance. Initially, we set  $k=1$  to assess the model's performance and then used the elbow method, which involves plotting accuracy against various  $k$  values to identify where increasing  $k$  no longer significantly improves accuracy. Using this method, we found that the best accuracy is achieved when  $k=39$ .

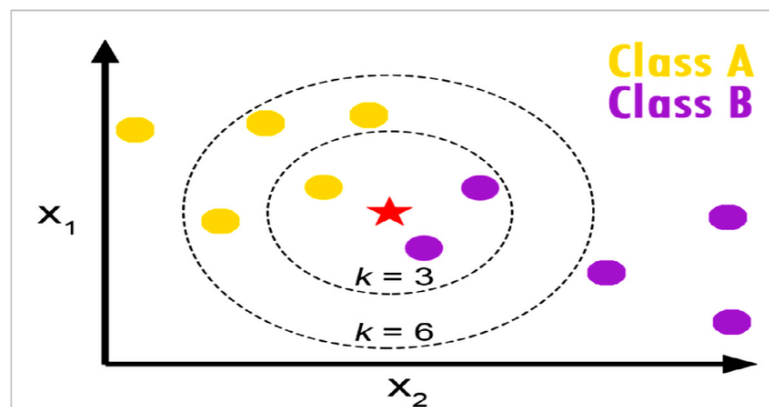


FIGURE 6 : SCHEMATIC REPRESENTATION OF K-NEAREST NEIGHBOUR (KNN) ALGORITHM CONCEPT. WHERE,  $k$  = NUMBER OF NEIGHBOURS;  $x_1$  &  $x_2$  ARE THE INPUT DESCRIPTORS; RED STAR IS THE QUERY INSTANCE.

#### 5) Random Forest

Random Forest is an ensemble learning method that enhances classification accuracy and mitigates overfitting by combining multiple decision trees. Each tree is trained on a random subset of both the data and features. The final prediction is then made by aggregating the predictions of all trees, usually through majority voting. Decision trees serve as the fundamental components of a Random Forest, partitioning the data into subsets based on feature values and making decisions to classify them. While individual trees may suffer from overfitting, their collective aggregation in a Random Forest results in a robust model with enhanced generalization capabilities to new data.

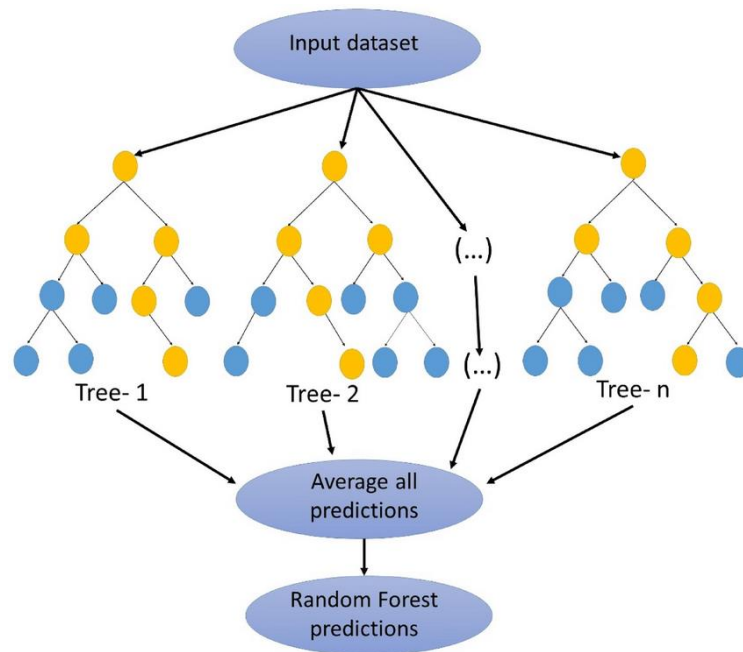


FIGURE 7 : SCHEMATIC DIAGRAM OF RANDOM FOREST CLASSIFIER

### 3.2.5 Deep Learning approaches

#### 1) Recurrent Neural Network

Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other. Still, in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is its Hidden state, which remembers some information about a sequence. The state is also referred to as Memory State since it remembers the previous input to the network. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural network

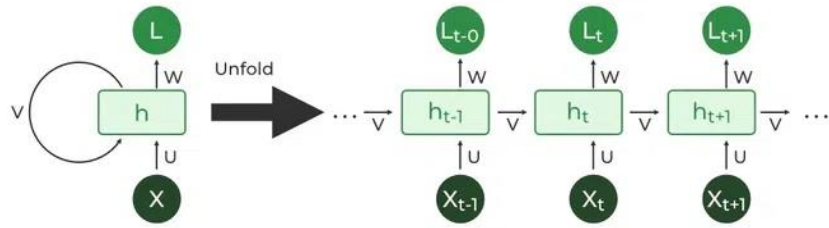


FIGURE 8 : RECURRENT NEURAL NETWORK

## 2) Gated Recurrent Unit

Gated Recurrent Unit (GRU), introduced as a simpler alternative to Long Short-Term Memory (LSTM) networks by Cho et al. in 2014, is a type of recurrent neural network (RNN) capable of processing sequential data like text, speech, and time-series data. GRU utilizes gating mechanisms to selectively update the hidden state at each time step, controlling the flow of information in

and out of the network. It employs two gating mechanisms: the reset gate, which determines how much of the previous hidden state to forget, and the update gate, which determines how much of the new input to use for updating the hidden state. The output is calculated based on the updated hidden state, with equations defining the reset gate, update gate, and hidden state. These gates address the vanishing gradient problem of standard RNNs by selectively retaining or discarding information over time, enhancing the network's ability to capture long-term dependencies and relevant features for prediction.

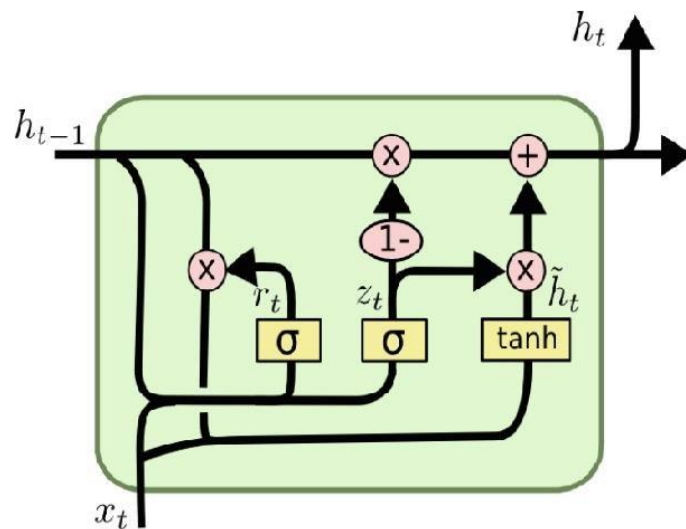


FIGURE 9 : GATED NEURAL NETWORK

### 3) Long Short Term Memory

Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) designed to address the vanishing gradient problem in traditional RNNs. Its advantage lies in its ability to capture long-term dependencies in sequential data, such as text, due to its unique architecture and gated mechanisms. LSTM includes input, forget, and output gates that control the flow of information, allowing it to selectively update and maintain a cell state over time. This capability enables LSTM to retain relevant information across thousands of timesteps, making it effective for various tasks like text classification, language modeling, speech recognition, and healthcare applications.

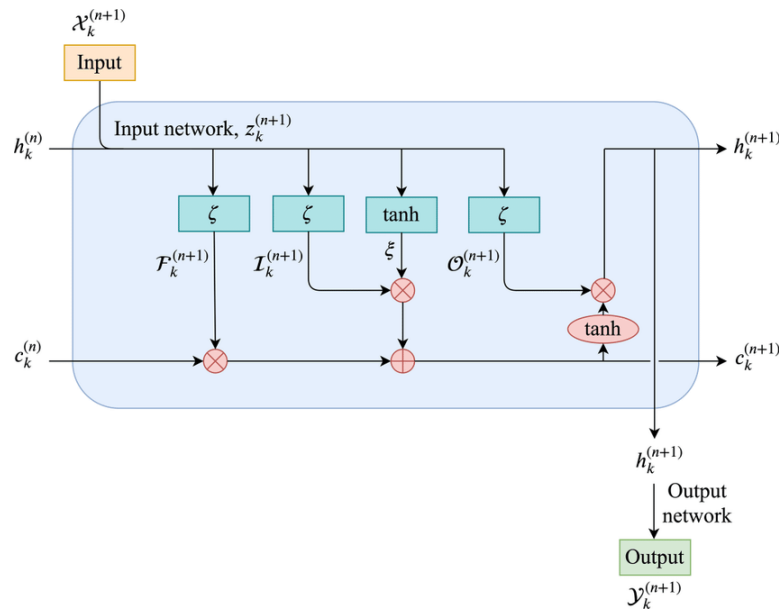


FIGURE 10 : LONG SHORT TERM MEMORY

### 4) Bidirectional-LSTM

Bidirectional LSTM (BiLSTM) is a sequence model comprising two LSTM layers, processing input in both forward and backward directions, commonly employed in NLP tasks. By analyzing data bidirectionally, BiLSTM captures relationships between sequences more effectively than unidirectional LSTM, enhancing its performance in tasks like sentiment analysis, text classification, and machine translation. Its architecture consists of two separate LSTM networks, each processing the sequence in one direction, with final output derived from a combination of the probability vectors generated by both networks. This approach enables BiLSTM to better understand sequence dependencies and yield more accurate predictions.

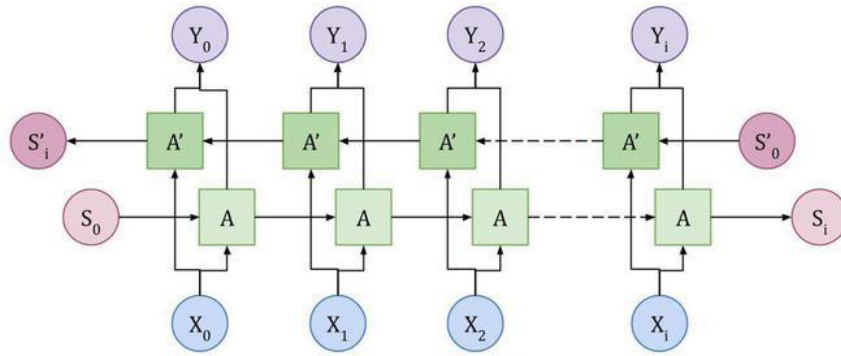


FIGURE 11 : DESCRIBES THE ARCHITECTURE OF THE BiLSTM LAYER WHERE  $X_i$  IS THE INPUT TOKEN,  $Y_i$  IS THE OUTPUT TOKEN, AND  $A$  AND  $A'$  ARE LSTM NODES. THE FINAL OUTPUT OF  $Y_i$  IS THE COMBINATION OF  $A$  AND  $A'$  LSTM NODES.

## Chapter 4 : Implementation and Results

---

### 4.1 Datasets

The dataset description of both the problem statements are shown below :

#### 4.1.1 Dataset for problem statement 1 :

This dataset is of Danish parliamentary debate and it is taken from the website – “Touche at CLEF 2024 – Webis Group” . In this website it’s mentioned that the data comes from [ParlaMint](#), a multilingual comparable corpora of parliamentary debates.

The dataset contains following columns :

- **id** is a unique (arbitrary) ID for each text.
- **speaker** is a unique (arbitrary) ID for each speaker. There may be multiple speeches from the same speaker.
- **sex** is the (binary/biological) sex of the speaker. The values in this field can be Female, Male, and Unspecified/Unknown.
- **text** is the transcribed text of the parliamentary speech. Real examples may include line breaks, and other special sequences escaped or quoted.
- **text\_en** is the automatic translation of the text to English. This field may be empty - obviously for speeches in English, but the translation may also be missing for a small number of non-English speeches.
- **label** is the binary/numeric label. For power identification 1 indicates opposition and 0 indicates coalition (or governing party).

**Size of dataset : 5493 rows and 6 columns**

**Number of sentences per label:**

Number of sentences for label 0 is 55569

Number of sentences for label 1 is 68830

**Number of words per label:**

Number of words for label 0 is 1433180

Number of words for label 1 is 1751543

**Total number of 0s and 1s in label column are 2043 and 3450 respectively.**

### 4.1.2 Dataset for problem statement 2

The Parliament Digital Library stores pdf files of Lok Sabha Debate. Since these files are available in unstructured format, we have to pre-process it. First we have to convert the pdf to jpg files using a pdf to jpg converter, then we will get a folder containing images of all the pages of the pdf. This folder is needed to be uploaded in the google drive first, after uploading, we have to open the folder and then right click on the 1<sup>st</sup> image file and then click on "open with", and then select option "Google docs".

This will open the image file in Google docs and the extracted text from the image will be available there, we have to copy the text and then paste it in a word file. Ensure that the copied text contains only speaker name and speaker's speech (in the given format → speaker name : speaker's speech), any texts other than this if present should be removed. Repeat this steps for all the images present in the folder. Convert the final word file into a pdf file.

A python script which is designed which takes this pdf file as input, this script will give the desired dataset as output in form of an excel sheet.

The dataset contains following columns :

- **text\_id** is a unique (arbitrary) ID for each text. It starts from 100.
- **speaker\_id** is a unique (arbitrary) ID for each speaker. There may be multiple speeches from the same speaker so the speaker\_id field can have duplicate values. It starts from 1.
- **gender** is the gender of the speaker. The values in this field can be Female, Male, and Unspecified/Unknown.
- **speaker\_name** is the name of the speaker
- **text\_hi** is the parliamentary speech of speaker in Hindi language.
- **text\_en** is the parliamentary speech of speaker in English language.
- **label** can be 0, 1 or 2. For power identification 0 indicates opposition party, 1 indicates coalition (or governing party) and 2 indicates neither governing nor ruling (for example the speech of speaker of Lok Sabha is considered as neutral).

**Size of dataset : 220 rows and 7 columns**

**Number of sentences per label:**

Number of sentences for label 0 is 638

Number of sentences for label 1 is 1033

Number of sentences for label 2 is 217

**Number of words per label:**

Number of words for label 0 is 12272

Number of words for label 1 is 20649

Number of words for label 2 is 2847

Total number of 0s, 1s and 2s in label column are 58, 77 and 85 respectively.

## 4.2 Experiments and Results

The experimental outcomes of our report are shown below :

### Result of Machine Learning Models :

We have shown the precision, recall, f1-score, accuracy of five Machine Learning models on **Danish parliamentary debate dataset in Table 1**, here the highest score is of Support Vector Machine (accuracy = 74%) and lowest score is of Gaussian Naïve Bayes (accuracy = 53%).

| Model                                                                                                                             | Precision |      | Recall |      | F1-Score |      | Accuracy |
|-----------------------------------------------------------------------------------------------------------------------------------|-----------|------|--------|------|----------|------|----------|
|                                                                                                                                   | 0         | 1    | 0      | 1    | 0        | 1    |          |
| LR                                                                                                                                | 0.72      | 0.73 | 0.43   | 0.91 | 0.54     | 0.81 | 0.73     |
| SVM                                                                                                                               | 0.68      | 0.77 | 0.55   | 0.85 | 0.61     | 0.81 | 0.74     |
| GNB                                                                                                                               | 0.40      | 0.67 | 0.54   | 0.53 | 0.46     | 0.59 | 0.53     |
| KNN (K=1)                                                                                                                         | 0.44      | 0.69 | 0.52   | 0.62 | 0.47     | 0.65 | 0.58     |
| KNN (K=)                                                                                                                          | 0.52      | 0.68 | 0.33   | 0.83 | 0.40     | 0.75 | 0.65     |
| RF                                                                                                                                | 0.76      | 0.70 | 0.30   | 0.95 | 0.43     | 0.81 | 0.71     |
| LR - Logistic Regression, SVM - Support Vector Machine, GNB - Gaussian Naïve Bayes, KNN - K-Nearest Neighbour, RF - Random Forest |           |      |        |      |          |      |          |

TABLE 1 : RESULT OF MACHINE LEARNING MODELS ON DANISH DATASET

We have shown the precision, recall, f1-score, accuracy of five Machine Learning models on **Indian parliamentary debate dataset in Table 2**, here the highest score is of K-Nearest Neighbour (accuracy = 64%) and lowest score is of Gaussian Naïve Bayes and K-Nearest Neighbour with K=1 (accuracy = 41%).

| Model                                                                                                                            | Precision |      |      | Recall |      |      | F1-Score |      |      | Accuracy |
|----------------------------------------------------------------------------------------------------------------------------------|-----------|------|------|--------|------|------|----------|------|------|----------|
|                                                                                                                                  | 0         | 1    | 2    | 0      | 1    | 2    | 0        | 1    | 2    |          |
| LR                                                                                                                               | 1.00      | 0.50 | 0.55 | 0.14   | 0.71 | 0.75 | 0.25     | 0.59 | 0.63 | 0.55     |
| SVM                                                                                                                              | 0.60      | 0.50 | 0.58 | 0.21   | 0.71 | 0.69 | 0.32     | 0.59 | 0.63 | 0.55     |
| GNB                                                                                                                              | 0.27      | 0.36 | 0.64 | 0.21   | 0.57 | 0.44 | 0.24     | 0.44 | 0.52 | 0.41     |
| KNN (K=1)                                                                                                                        | 0.36      | 0.00 | 0.80 | 1.00   | 0.00 | 0.25 | 0.53     | 0.00 | 0.38 | 0.41     |
| KNN (K=8)                                                                                                                        | 0.56      | 0.62 | 0.73 | 0.64   | 0.57 | 0.69 | 0.60     | 0.59 | 0.71 | 0.64     |
| RF                                                                                                                               | 0.55      | 0.50 | 0.63 | 0.43   | 0.50 | 0.75 | 0.48     | 0.50 | 0.69 | 0.57     |
| LR - Logistic Regression, SVM - Support Vector Machine, GNB - Gaussian Naïve Bayes, KNN - K-Nearest Neighbor, RF - Random Forest |           |      |      |        |      |      |          |      |      |          |

TABLE 2 : RESULT OF MACHINE LEARNING MODELS ON INDIAN DATASET

## Result of Deep Learning Models

We have shown the precision, recall, f1-score, accuracy of four Deep Learning models on **Danish parliamentary debate dataset in Table 3**, here the highest score is of Bi-LSTM (accuracy = 67%) and lowest score is of LSTM (accuracy = 39%).

| Model                                                                                                                                              | Precision |      | Recall |      | F1-Score |      | Accuracy |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------|--------|------|----------|------|----------|
|                                                                                                                                                    | 0         | 1    | 0      | 1    | 0        | 1    |          |
| S-RNN                                                                                                                                              | 0.40      | 0.62 | 0.22   | 0.79 | 0.28     | 0.69 | 0.57     |
| GRU                                                                                                                                                | 0.50      | 0.68 | 0.49   | 0.69 | 0.50     | 0.68 | 0.61     |
| LSTM                                                                                                                                               | 0.39      | 0.00 | 1.00   | 0.00 | 0.56     | 0.00 | 0.39     |
| Bi-LSTM                                                                                                                                            | 0.60      | 0.71 | 0.48   | 0.80 | 0.53     | 0.75 | 0.67     |
| S-RNN – Simple Recurrent Neural Network, GRU – Gated Recurrent Unit, LSTM – Long Short Term Memory, Bi-LSTM – Bidirectional Long Short Term Memory |           |      |        |      |          |      |          |

TABLE 3 : RESULT OF DEEP LEARNING MODELS ON DANISH DATASET

## 4.3 Error Analysis

The Confusion matrix for five Machine Learning models that we have used are shown below in context of Danish parliamentary debate dataset :

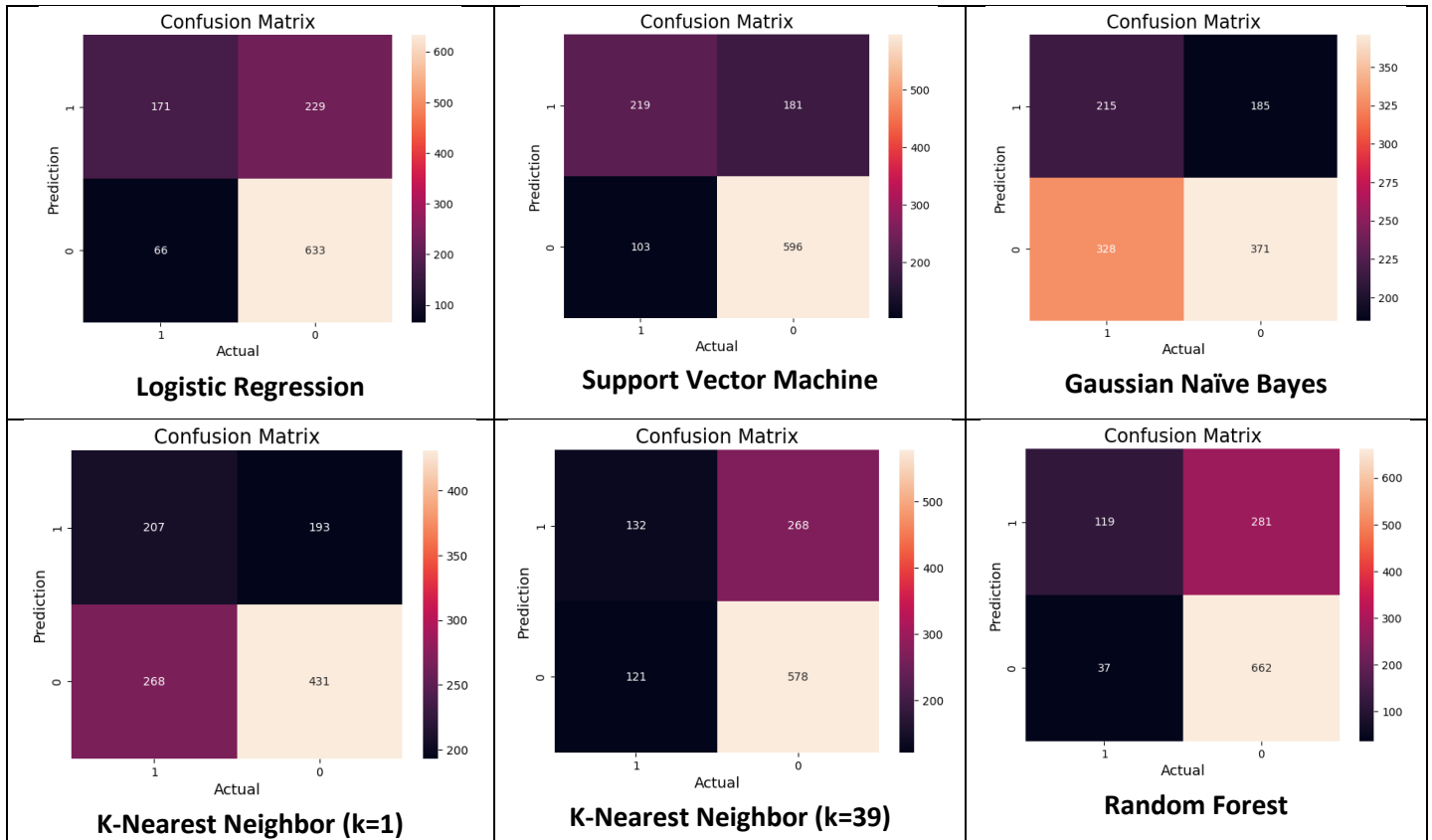


TABLE 4 : CONFUSION MATRICES FROM DANISH PARLIAMENTARY DEBATE DATASET CLASSIFICATION USING MACHINE LEARNING

With the help of Elbow method, we found that the error rate is lowest when value of K=39, as shown in Figure 10.

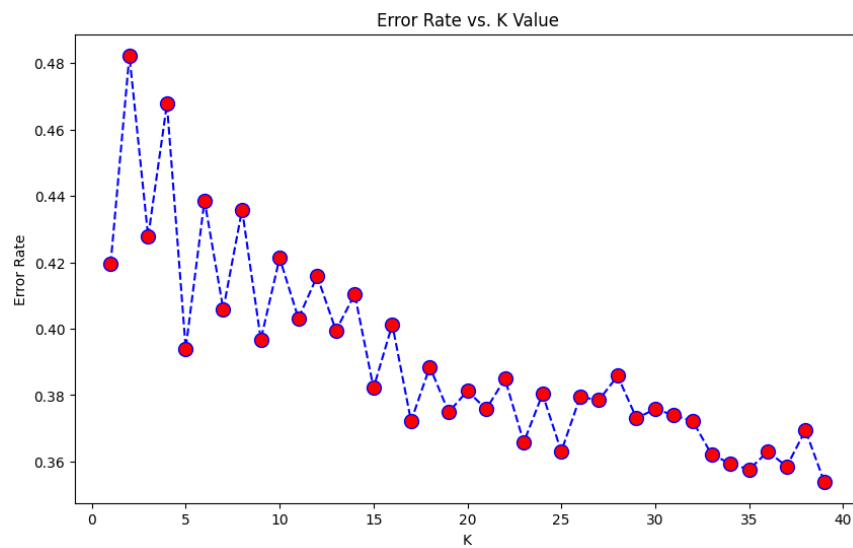
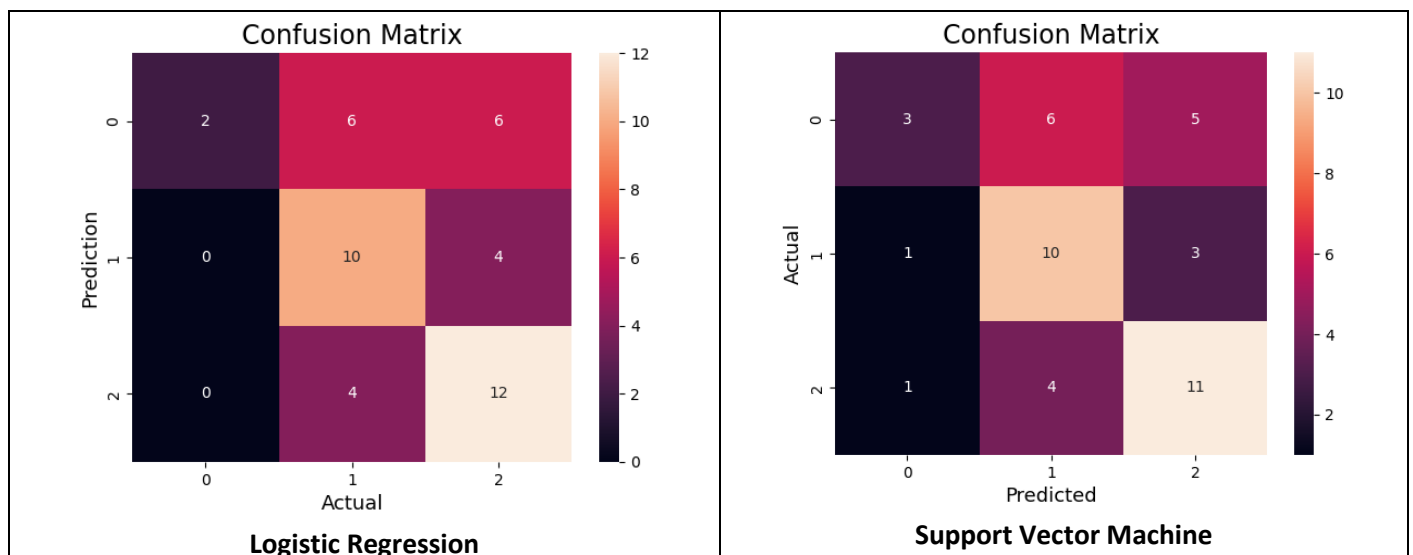


FIGURE 12 : ELBOW METHOD USED IN DANISH PARLIAMENTARY DEBATE DATASET CLASSIFICATION

The Confusion matrix for five Machine Learning models that we have used are shown below in context of Indian parliamentary debate dataset :



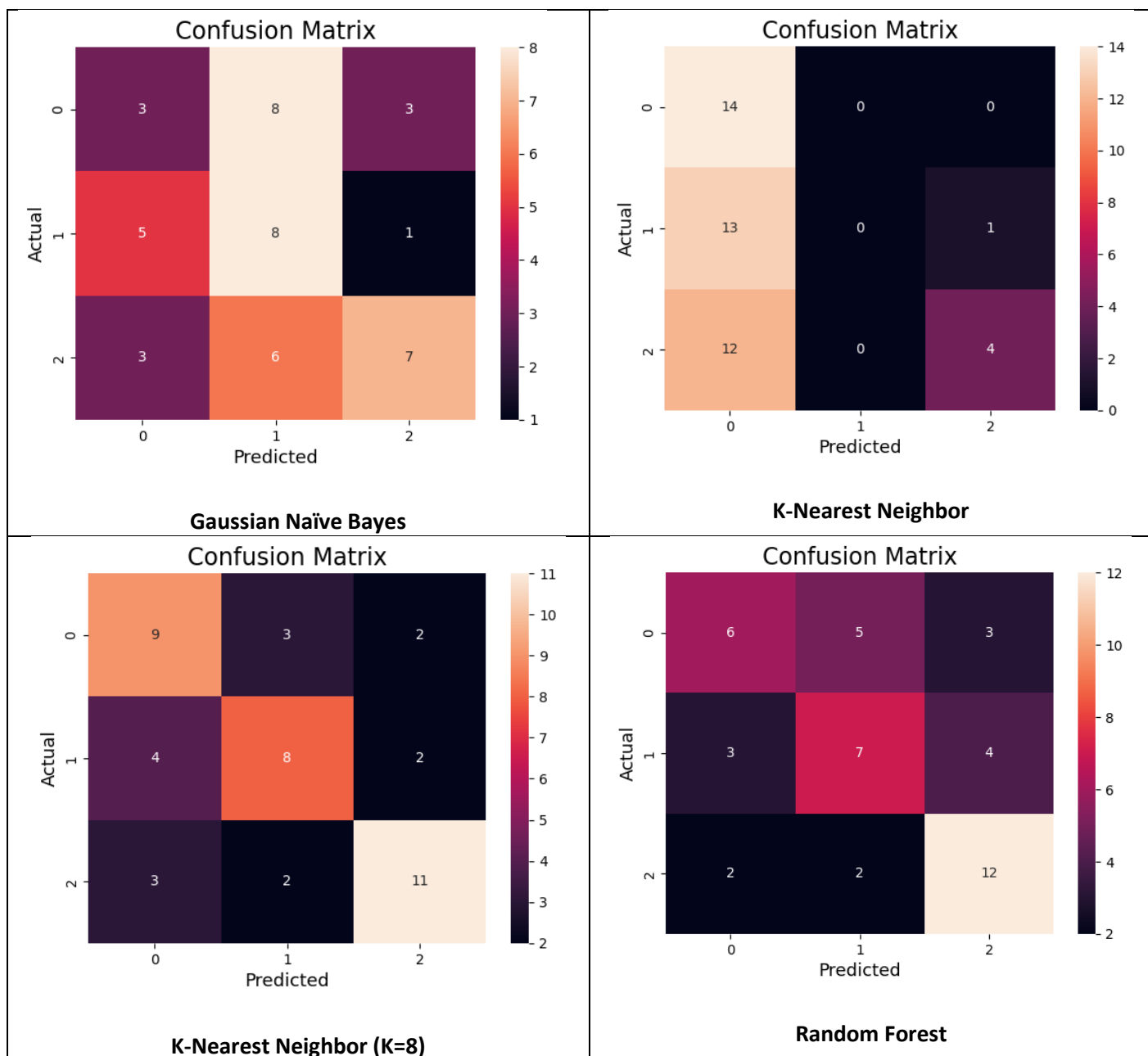


TABLE 5 : CONFUSION MATRICES FROM INDIAN PARLIAMENTARY DEBATE DATASET CLASSIFICATION USING MACHINE LEARNING

With the help of Elbow method, we found that the error rate is lowest when value of K=8, as shown in Figure 10.

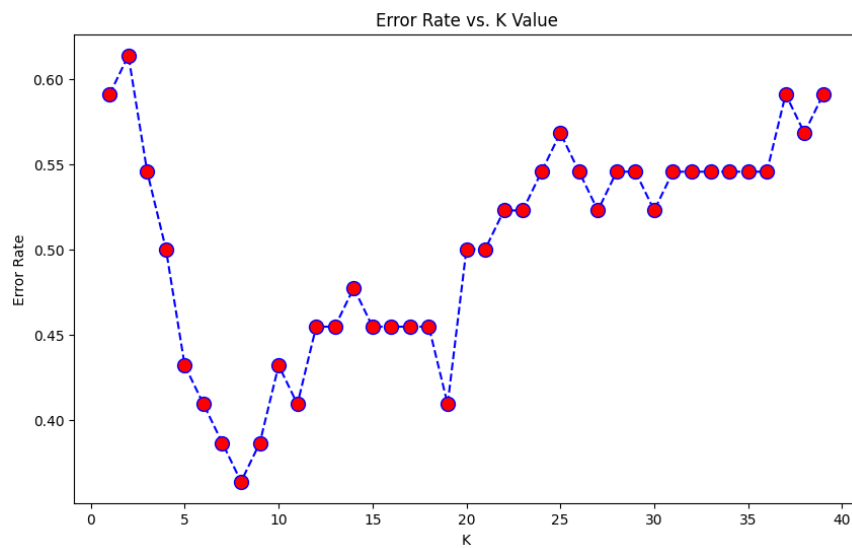
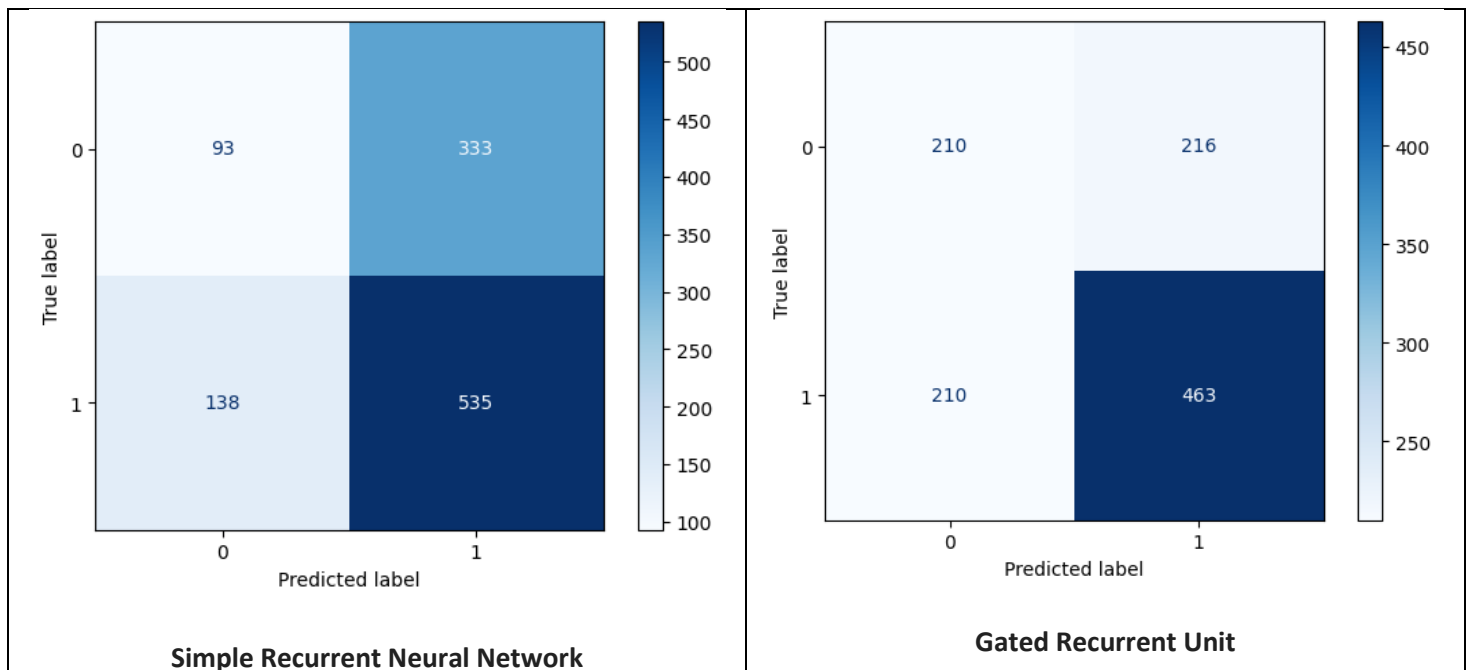


FIGURE 13 : ELBOW METHOD USED IN INDIAN PARLIAMENTARY DEBATE DATASET CLASSIFICATION

The Confusion matrix for four Deep Learning models that we have used are shown below in context of Danish parliamentary debate dataset :



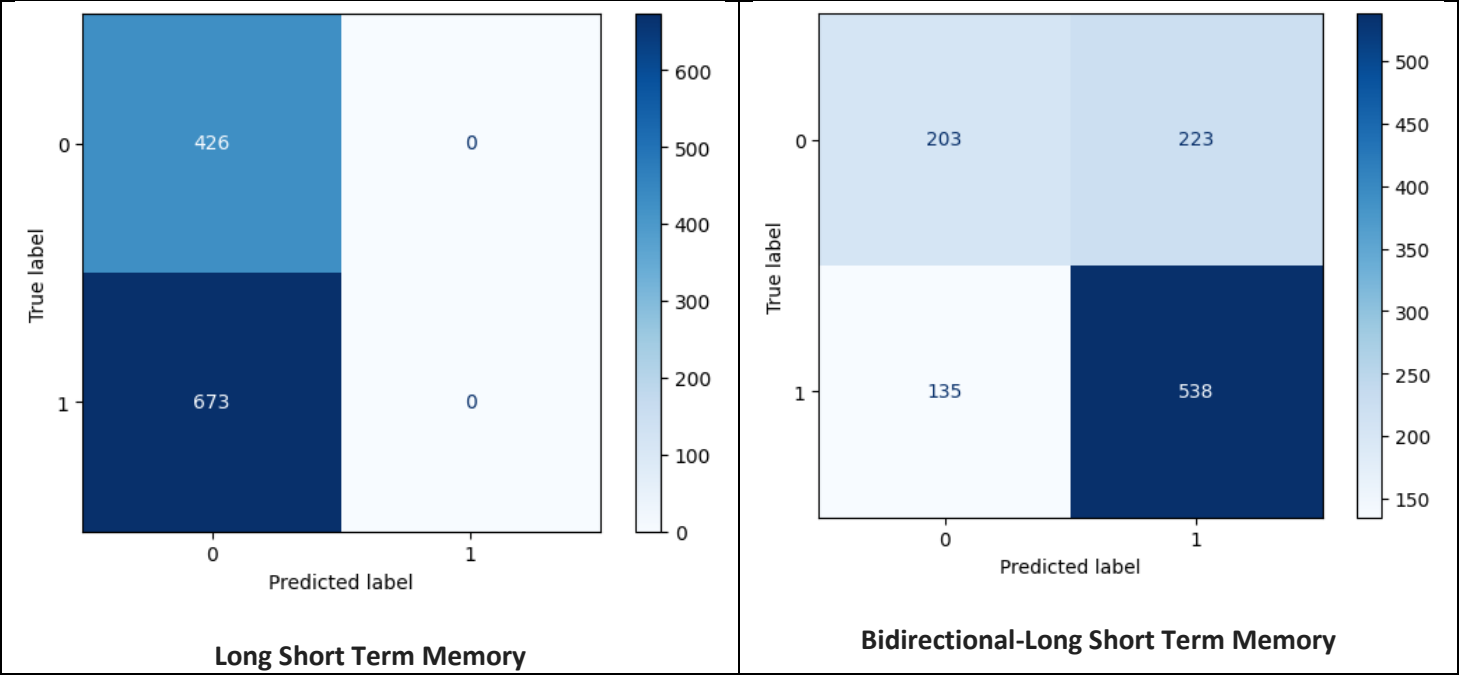


TABLE 6 : CONFUSION MATRICES FROM DANISH PARLIAMENTARY DEBATE DATASET CLASSIFICATION USING DEEP LEARNING

## Chapter 5 : Conclusion and Future Work

---

Expanding the dataset of Indian parliamentary debates, meticulously curated by our team, stands as a pivotal next step, recognizing its current size limitations (comprising 220 rows). This expansion will facilitate the application of more sophisticated deep learning models, given the inherent challenges such models face when operating on small datasets. Additionally, our pursuit extends to refining the accuracy achieved on the Danish parliamentary debate dataset, currently standing at 74% through Support Vector Machine utilization. Embracing alternative methodologies and approaches promises to unlock further enhancements in accuracy, thereby advancing the depth and precision of our analysis in parliamentary discourse classification.

Moving forward, our focus will encompass the integration of lexicon-based approaches and debate graph extraction frameworks into our analytical arsenal. Leveraging lexicon-based sentiment analysis techniques will enrich our understanding of the emotional undertones embedded within parliamentary speeches, complementing our existing machine learning methodologies. Furthermore, the adoption of debate graph extraction frameworks will empower us to unravel intricate relationships and patterns inherent in parliamentary discourse, thus enabling a more nuanced analysis. By integrating these innovative approaches, we aim to not only elevate the accuracy of our classification models but also deepen our insights into the dynamics of parliamentary debates.

## References

---

- [1] Salah, Zaher. "Machine learning and sentiment analysis approaches for the analysis of Parliamentary debates." PhD diss., University of Liverpool, 2014.
- [2] Chandra, Yogesh, and Antoreep Jana. "Sentiment analysis using machine learning and deep learning." In *2020 7th international conference on computing for sustainable global development (INDIACom)*, pp. 1-4. IEEE, 2020.
- [3] Umarani, V., Anitha Julian, and J. Deepa. "Sentiment analysis using various machine learning and deep learning Techniques." *Journal of the Nigerian Society of Physical Sciences* (2021): 385-394.
- [4] Lai, Mirko, Cristina Bosco, Viviana Patti, and Daniela Virone. "Debate on political reforms in Twitter: A hashtag-driven analysis of political polarization." In *2015 IEEE international conference on data science and advanced analytics (Dsa)*, pp. 1-9. IEEE, 2015.
- [5] Gupta, Shailja, Sachin Lakra, and Manpreet Kaur. "Sentiment analysis using partial textual entailment." In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 51-55. IEEE, 2019.
- [6] Rheault, Ludovic, Kaspar Beelen, Christopher Cochrane, and Graeme Hirst. "Measuring emotion in parliamentary debates with automated textual analysis." *PloS one* 11, no. 12 (2016): e0168843.
- [7] Rudkowsky, Elena, Martin Haselmayer, Matthias Wastian, Marcelo Jenny, Štefan Emrich, and Michael Sedlmair. "More than bags of words: Sentiment analysis with word embeddings." *Communication Methods and Measures* 12, no. 2-3 (2018): 140-157.
- [8] Proksch, Sven-Oliver, Will Lowe, Jens Wäckerle, and Stuart Soroka. "Multilingual sentiment analysis: A new approach to measuring conflict in legislative speeches." *Legislative Studies Quarterly* 44, no. 1 (2019): 97-131.
- [9] Ain, Qurat Tul, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat, and A. Rehman. "Sentiment analysis using deep learning techniques: a review." *International Journal of Advanced Computer Science and Applications* 8, no. 6 (2017).
- [10] Pandian, A. Pasumpon. "Performance evaluation and comparison using deep learning techniques in sentiment analysis." *Journal of Soft Computing Paradigm (JSCP)* 3, no. 02 (2021): 123-134.