# Indian Institute of Information Technology Surat


# Project Report on
## Machine Learning (CS 601) Practical

**Submitted by**

**[RAHUL KUMAR SINGH] (UI21CS44)**

**Course Faculty**

**Dr. Pradeep Kumar Roy**

**Department of Computer Science and Engineering**
**Indian Institute of Information Technology Surat**
**Gujarat-394190, India**

**Jan-2024**

# Table of Contents


I Introduction 4
- 1 Project Overview 4
- 2 The Purpose of the Project 4
  - 2a The Importance of Solving the Identified Problem 4
  - 2b Goals of the Project (UN Goals) 4
- 3 The Domain of the Project 5
  - 3a Methodology 5
- 4 Existing Solutions 5
- 5 Proposed Solution 6
  - 5a Model Evaluation Summary 6

II Existing Techniques and Solutions for Legal Prediction 6
- 1 Machine Learning-based SON Function Conflict Resolution (Reference: [1]) 6
  - 1a Techniques Used 7
  - 1b Key Findings 7
- 2 Conflict Resolution Strategies in AI (Reference: [2]) 7
  - 2a Techniques Used 7
  - 2b Key Findings 7
- 3 Artificial Intelligence Techniques for Conflict Resolution (Reference: [3]) 8
  - 3a Techniques Used 8
  - 3b Key Findings 8
- 4 Using Artificial Intelligence to Provide Intelligent Dispute Resolution (Reference: [4]) 8
  - 4a Techniques Used 8
  - 4b Key Findings 9

III Proposed System 9
- 1 Dataset 9
- 2 Modules 9
- 3 Models 10
  - 3a Model Overviews 10
- 4 Experiments 10
  - 4a Data Preprocessing 11
  - 4b Data Anonymization 11
  - 4c Label Class Imbalance 12
- 5 Training 12
- 6 Final Steps 13
  - 6a Ensemble Learning 13
  - 6b Model Selection 14
  - 6c Deployment 14

IV Results 14
- 1 Model/Label Selection 14

1a TF-IDF 14
1b GloVe 15
1c Doc2Vec 15
1d CNN 15
2 Confusion Matrix 16
2a TF-IDF 16
2b GloVe 16
2c Doc2Vec 17
2d CNN 17
2e LSTM 18
2f BERT 18
2g Ensemble 18
V Summary 19
1 TF-IDF 19
2 GloVe 19
3 Doc2Vec 19
4 CNN: 19
VI References / Bibliography 20

# I Introduction

## 1 Project Overview

In today's legal landscape, the ability to predict Supreme Court decisions accurately holds immense significance. We introduce lawGov, a groundbreaking Natural Language Processing (NLP) application designed to predict the ideological direction of Supreme Court case decisions. Our goal is to provide valuable insights to parties involved in upcoming cases, enabling them to assess their prospects for success and plan their litigation strategies accordingly.

## 2 The Purpose of the Project

### 2a The Importance of Solving the Identified Problem

The ability to predict legal judgments with precision is crucial for several reasons. First and foremost, it enables legal professionals to make informed decisions and devise effective strategies for their clients. By having insights into potential outcomes, lawyers can better advise their clients, manage expectations, and optimize resources.

Furthermore, accurate legal predictions can enhance access to justice. Many individuals and businesses are deterred from pursuing legal action due to uncertainty regarding the outcome and the associated costs. lawGov can help mitigate these concerns by providing clarity on the likely outcome of a case, thereby promoting fairness and transparency within the legal system.

Moreover, the efficiency gains offered by lawGov are significant. Legal research and analysis are time-consuming tasks that often involve sifting through mountains of documents. By automating this process and providing accurate predictions, lawGov enables legal professionals to focus their time and energy on more strategic aspects of their work, ultimately improving productivity and reducing costs.

### 2b Goals of the Project (UN Goals)

Goal 16: Peace, Justice, and Strong Institutions

Target 16.6: Develop effective, accountable, and transparent institutions at all levels.

Goal 9: Industry, Innovation, and Infrastructure

Target 9.5: Enhance scientific research, upgrade technological capabilities, and promote innovation.

Goal 10: Reduced Inequalities

Target 10.2: Empower and promote the social, economic, and political inclusion of all, irrespective of age, sex, disability, race, ethnicity, origin, religion, or economic or other status.

# 3 The Domain of the Project

The legal domain is complex, multifaceted, and often fraught with uncertainty. Legal professionals face the daunting task of analyzing vast amounts of information, including past cases, legal precedents, statutes, and regulations, to make informed decisions. However, even the most seasoned legal experts can struggle with predicting the outcome of a case accurately. This is where lawGov steps in, utilizing advanced NLP algorithms to analyze and interpret legal data swiftly and accurately.

## 3a Methodology

Modeling the Court as a Whole:

This approach involves predicting the single outcome of a case based on the generally nine justices' votes.

Ensemble Method:

Here, we separately model the decisions of individual justices in probability terms and then aggregate these probabilities to predict the outcome of the case.

# 4 Existing Solutions

Existing techniques and solutions for legal prediction encompass various approaches, including machine learning (ML) and artificial intelligence (AI) methodologies. These techniques leverage advanced algorithms to improve legal decision-making, with ML achieving up to 90% prediction accuracy (theoretical and case study), and AI-based systems providing automated mediation and early warning systems for conflicts.

Machine Learning-based SON Function Conflict Resolution

Sarker et al. used reinforcement learning and clustering algorithms to dynamically optimize SON functions, enhancing network performance and resource allocation.

Conflict Resolution Strategies in AI

Game theory and multi-agent systems are used to model and analyze conflicts, aiding in strategic decision-making and resource optimization.

Artificial Intelligence Techniques for Conflict Resolution

Noorian et al. employ neural networks and fuzzy logic, enabling adaptive decision-making and efficiency gains.

Using Artificial Intelligence to Provide Intelligent Dispute Resolution

Lin et al. apply NLP and case-based reasoning, resulting in automated mediation and consistent, fair outcomes.

## 5 Proposed Solution

We conducted 8 key experiments on a dataset. Data preprocessing included removing stopwords, lowercasing, stemming, and cleaning non-alphabet characters. Anonymization replaced party names with _PARTY_, and class imbalance was addressed. With an 80:20 train-test split, we used 4-fold cross-validation. After training, we achieved an average testing accuracy of 85%.

### 5a Model Evaluation Summary

TF-IDF:

Among combinations 3 and 4, the fourth model of the third combination achieved the highest testing accuracy of 0.972 and a testing loss of 0.141.

GloVe:

Combination 2 yielded the best results, with the first model achieving a testing accuracy of 0.916 and a testing loss of 0.384.

Doc2Vec:

The second model of combination 5 outperformed others, achieving a testing accuracy of 0.945 and a testing loss of 0.282.

CNN:

For combinations 2 and 5, the second model of the second combination had the highest testing accuracy of 0.933 and a testing loss of 0.325.

Overall, TF-IDF with combination 3's fourth model demonstrated the best performance, achieving the highest testing accuracy of 0.972, while GloVe's best model in combination 2 came close with an accuracy of 0.916. Doc2Vec's top model and CNN's top model also showed competitive performance, with accuracies of 0.945 and 0.933, respectively.

# II Existing Techniques and Solutions for Legal Prediction

## 1 Machine Learning-based SON Function Conflict Resolution (Reference: [1])

The study by Sarker et al. focuses on using machine learning techniques for resolving conflicts in Self-Organizing Network (SON) functions, with implications for telecommunications and beyond.

### 1a Techniques Used

Reinforcement Learning

Sarker et al. employ reinforcement learning algorithms to optimize SON function allocation and resolve conflicts between network elements.

Clustering Algorithms

Clustering techniques are used to group network elements based on their characteristics, facilitating conflict resolution strategies.

### 1b Key Findings

Dynamic Optimization

ML-based conflict resolution enables dynamic optimization of SON functions, adapting to changing network conditions in real-time.

Resource Allocation

By efficiently allocating resources and mitigating conflicts, ML models improve network performance and reliability.

## 2 Conflict Resolution Strategies in AI (Reference: [2])

The article on Conflict Resolution Strategies in Artificial Intelligence explores various AI-based approaches to resolving conflicts, including those encountered in legal contexts.

### 2a Techniques Used

Game Theory

Game theoretic approaches are employed to model and analyze conflicts, considering the strategic interactions between parties involved.

Multi-Agent Systems

AI techniques such as multi-agent systems are utilized to simulate and study complex interactions between legal entities, facilitating conflict resolution.

### 2b Key Findings

Strategic Decision-Making

AI-based conflict resolution strategies enable strategic decision-making, allowing legal professionals to anticipate and respond to adversarial behavior.

Optimization

By identifying optimal strategies for conflict resolution, AI systems help streamline legal processes and reduce resource expenditure.

# 3 Artificial Intelligence Techniques for Conflict Resolution (Reference: [3])

The article by Noorian et al. investigates the application of AI techniques for conflict resolution, focusing on their utility in mitigating disputes and fostering cooperation.

## 3a Techniques Used

Neural Networks

Noorian et al. explore the use of neural networks for conflict resolution, leveraging their ability to learn and adapt from data.

Fuzzy Logic

Fuzzy logic-based systems are employed to model uncertainty and vagueness inherent in legal disputes, allowing for more nuanced decision-making.

## 3b Key Findings

Adaptive Decision-Making

AI techniques enable adaptive decision-making in response to changing circumstances, enhancing the agility of legal systems.

Efficiency Gains

By automating certain aspects of conflict resolution, AI systems contribute to efficiency gains and resource optimization.

# 4 Using Artificial Intelligence to Provide Intelligent Dispute Resolution (Reference: [4])

The article by Lin et al. discusses the use of AI to provide intelligent dispute resolution mechanisms, emphasizing their role in facilitating efficient and fair outcomes.

## 4a Techniques Used

Natural Language Processing (NLP)

Lin et al. leverage NLP techniques to analyze legal texts and extract relevant information, enabling automated dispute resolution.

Case-Based Reasoning

Case-based reasoning systems are utilized to identify similarities between current and past cases, guiding decision-making processes.

### 4b Key Findings

Automated Mediation

AI-based dispute resolution systems automate mediation processes, reducing the need for human intervention and expediting case resolution.

Fairness and Consistency

By applying consistent decision-making criteria, AI systems promote fairness and transparency in dispute resolution.

# III Proposed System

## 1 Dataset

The dataset comprises 23464 legal cases covering various fields. The key features include first_party, second_party, winner_index, and facts. Here is an overview of the dataset structure:

- ID (int64): Defines the case ID
- name (string): Defines the case name
- href (string): Defines the case hyper-reference
- first_party (string): Defines the name of the first party (petitioner) of a case
- second_party (string): Defines the name of the second party (respondent) of a case
- winning_party (string): Defines the winning party name of a case
- winner_index (int64): Defines the winning index of a case, 0 => the first party wins, 1 => the second party wins
- facts (string): Contains the case facts that are needed to determine who is the winner of a specific case

The input for lawGov models will be the facts, and the target will be the winner_index.

## 2 Modules

To maintain organization, the codebase is divided into 5 modules:

- Preprocessing Module: Responsible for preprocessing case facts including tokenization, balancing data, and anonymizing facts.
- Plotting Module: Manages plotting and visualization of performance measures of lawGov models.
- Utils Module: Contains reusable functions such as model training and evaluation.
- Main Module: Handles the deployment using Streamlit for the frontend.

- Deployment Utils Module: Contains functions and classes for model deployment and predictions.

# 3 Models

Each model has been selected to explore various techniques and their effectiveness in predicting legal case outcomes.

## 3a Model Overviews

- Doc2Vec: Captures document-level semantics.
- 1D-CNN: Learns features from raw textual data.
- TextVectorization with TF-IDF: Converts text data into numerical vectors using TF-IDF.
- GloVe: Generates word embeddings based on co-occurrence statistics.
- BERT: Utilizes a bidirectional transformer architecture for language understanding.
- LSTM: Overcomes the limitations of traditional RNNs in capturing long-term dependencies.
- FastText: Represents words using character n-grams.

# 4 Experiments

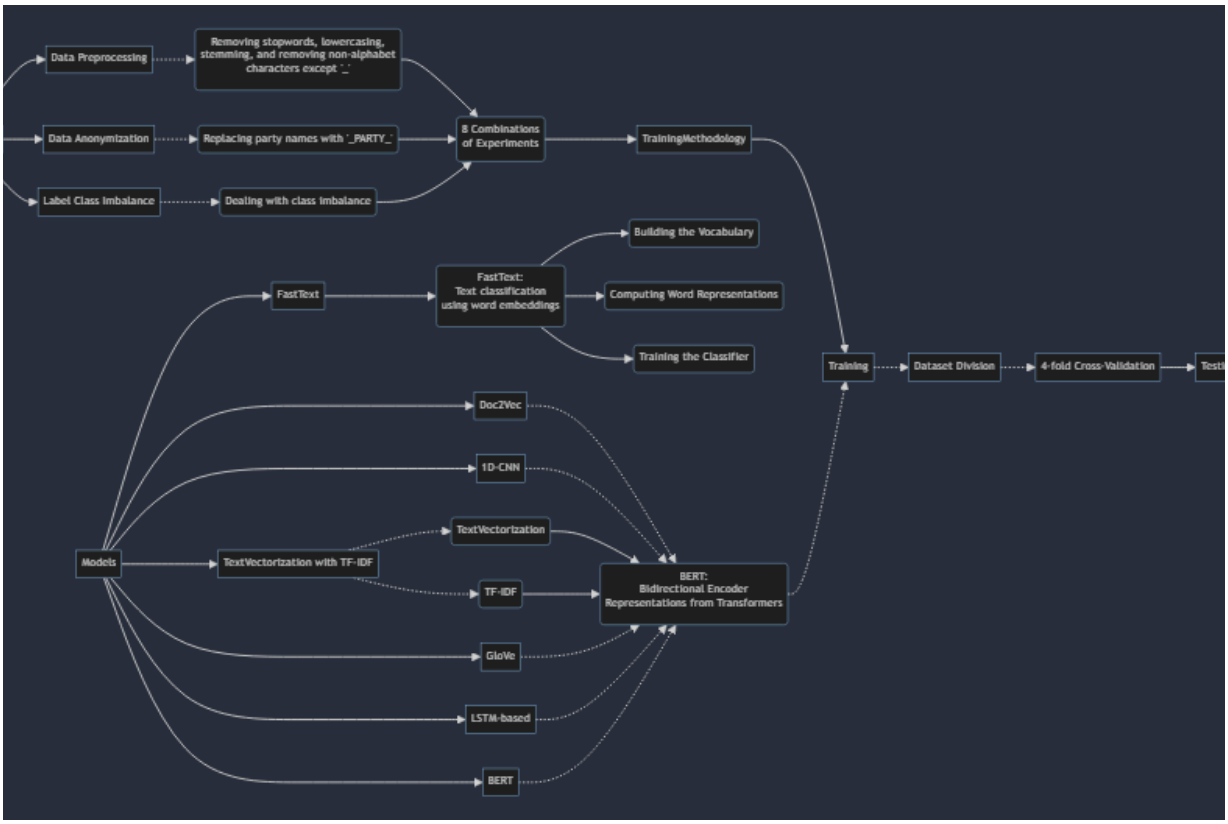


**Figure 1.1 - Model Architecture for lawGov Model (v1)**

To ensure the effectiveness of lawGov models, three key experiments were conducted

## 4a Data Preprocessing

Objective:

To optimize text data for better model performance.

Steps:

- Removing stopwords: Commonly used words that do not contribute to the meaning (e.g., "the", "and").
- Lowercasing: Converting all text to lowercase to ensure uniformity.
- Stemming: Reducing words to their root form (e.g., "running" to "run").
- Cleaning non-alphabet characters: Removing symbols, digits, and special characters except underscore _.

Impact:

This experiment aimed to enhance the quality of text data for model training.

## 4b Data Anonymization

Objective:

To reduce bias towards parties' names in case facts.

Steps:

Replacing parties' names with a generic tag (_PARTY_).

Impact:

This experiment aimed to prevent models from learning biases associated with specific party names, ensuring fairness in predictions.

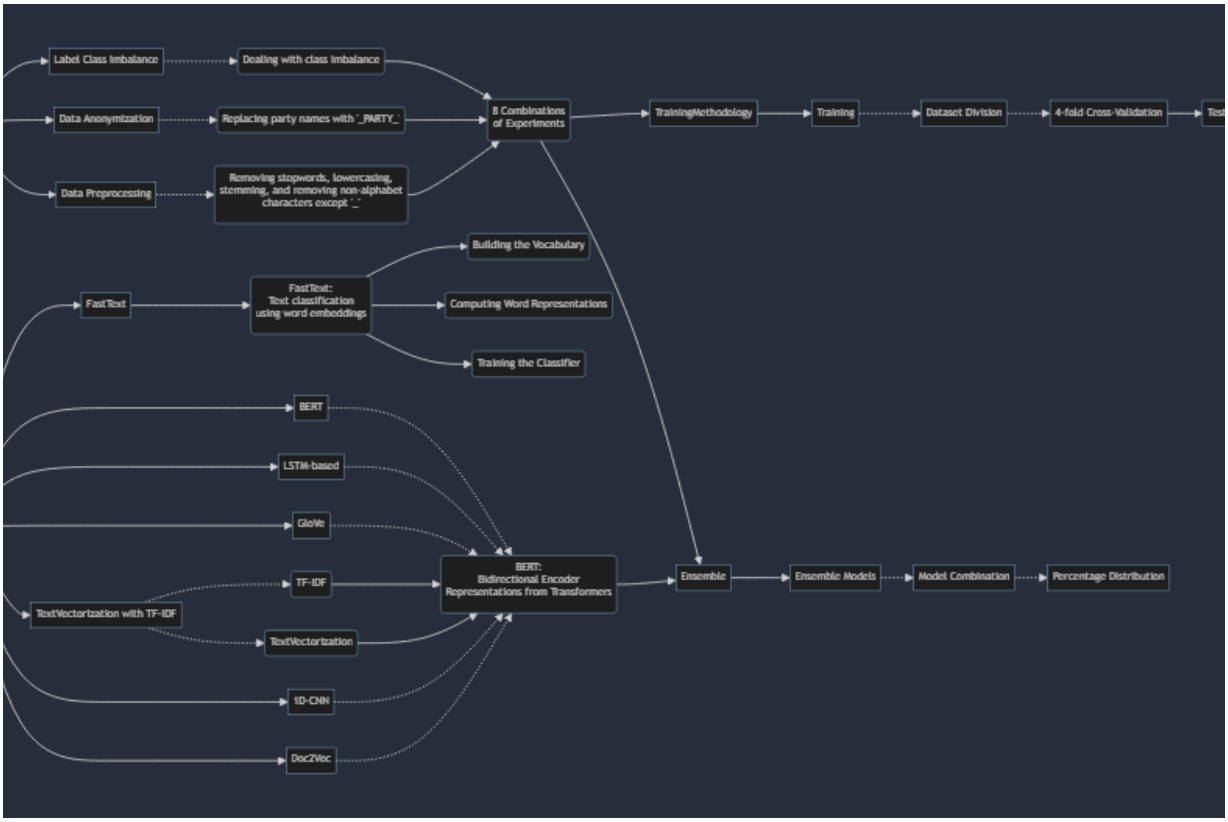


**Figure 1.2 - Model Architecture for lawGov Model (v9)**

### 4c Label Class Imbalance

Objective:

To handle imbalanced classes in the target variable.

Steps:

Balancing the number of cases where each party wins.

Impact:

Dealing with class imbalance aimed to improve the model's ability to predict outcomes for both parties equally.

## 5 Training

Data Splitting:

- The dataset was divided into 80% training and 20% testing data.
- This division remained constant across all models for consistent evaluation.

Cross-validation:

- Utilized 4-fold cross-validation for training each model.
- The training data was split into four equal parts or folds.

Evaluation:

- Model performance was evaluated using testing accuracy for each fold.
- Testing accuracies were averaged to assess overall model performance.

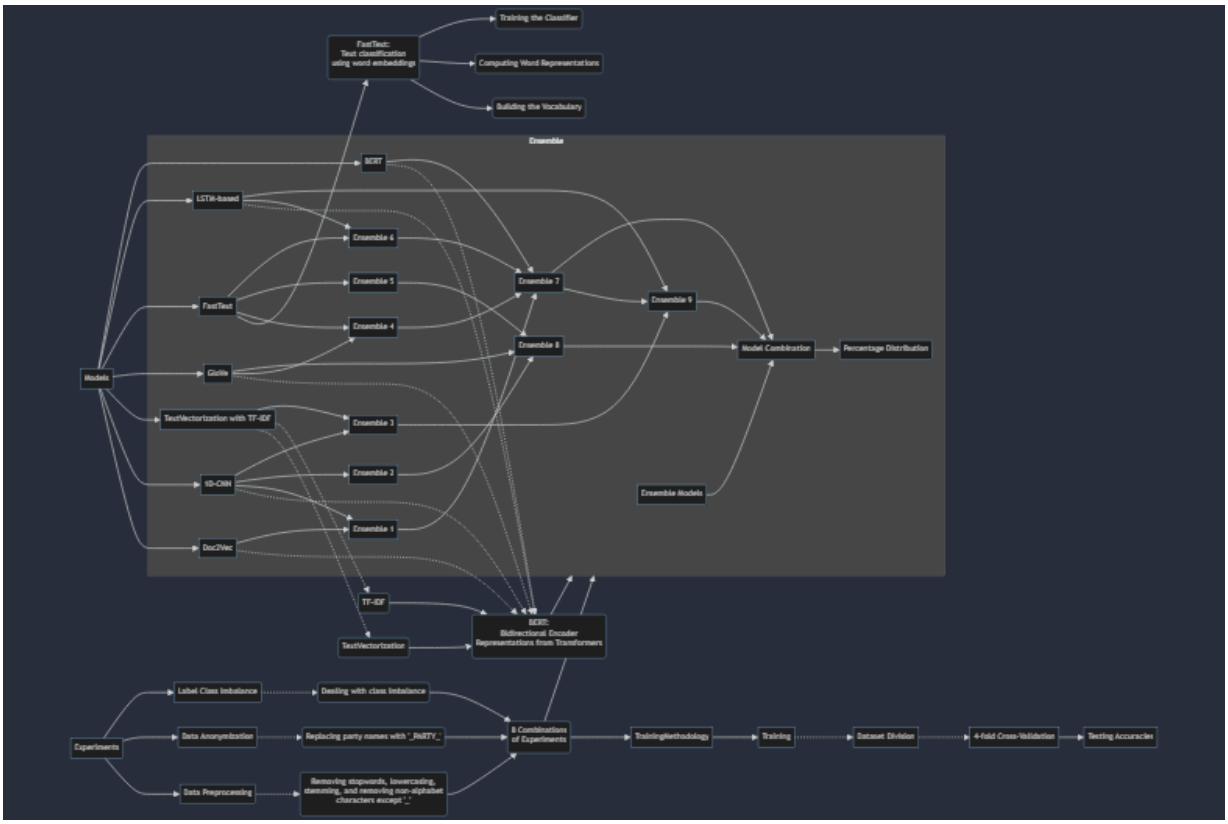


**Figure 1.3 - Model Architecture for lawGov Model (v23)**

# 6 Final Steps

After training and selecting the best-performing model combination, the following steps were taken:

## 6a Ensemble Learning

Objective:

To combine predictions from multiple models for improved accuracy.

Steps:

- Implemented a simple ensemble approach using voting.
- Each model votes on the winning party for a specific case.
- The winning party is determined by majority voting.

Impact:

Ensemble learning leveraged the strengths of different models, resulting in more accurate predictions.

#### 6b Model Selection

Objective:

Choose the best combination of model and preprocessing steps.

Criteria:

- High testing accuracy.
- Generalization on the testing set.

Selection Process:

- Identified the best combination based on testing accuracies from cross-validation.
- Saved the best-performing model combination for deployment.

#### 6c Deployment

Objective:

Deploy the best model combination for real-world use.

Steps:

- Utilized the selected model for deployment.
- Developed a user-friendly frontend using Streamlit for inputting case facts.
- Implemented prediction logic to provide accurate outcomes based on the input facts.

# IV Results

## 1 Model/Label Selection

### 1a TF-IDF

Among the combinations tested, two similar combinations emerged: combination 3 (no preprocessing - anonymization - imbalance) and combination 4 (no preprocessing - anonymization - balanced), each with four results based on folds. After evaluation, the model that exhibited the best generalization on testing data, achieving the highest

testing accuracy, is the fourth model of the third combination, with a testing accuracy of 0.972 and a testing loss of 0.141.

### 1b GloVe

The best performing combination is found in combination 2 (no preprocessing - no anonymization - balanced). After analysis, the model that demonstrated the highest testing accuracy and best generalization on testing data is the first model of the second combination, with a testing accuracy of 0.916 and a testing loss of 0.384.

### 1c Doc2Vec

Two similar combinations emerged from the evaluation: combination 1 (no preprocessing - no anonymization - imbalance) and combination 5 (preprocessing - no anonymization - imbalance), each with four results based on folds. The model showing the best generalization on testing data, with the highest testing accuracy, is the second model of the fifth combination, achieving a testing accuracy of 0.945 and a testing loss of 0.282.

### 1d CNN

Two similar combinations were identified: combination 2 (no preprocessing - no anonymization - balanced) and combination 5 (preprocessing - no anonymization - imbalance), each with four results based on folds. After evaluation, the model demonstrating the highest testing accuracy and best generalization on testing data is the second model of the second combination, with a testing accuracy of 0.933 and a testing loss of 0.325.

# 2 Confusion Matrix

## 2a TF-IDF

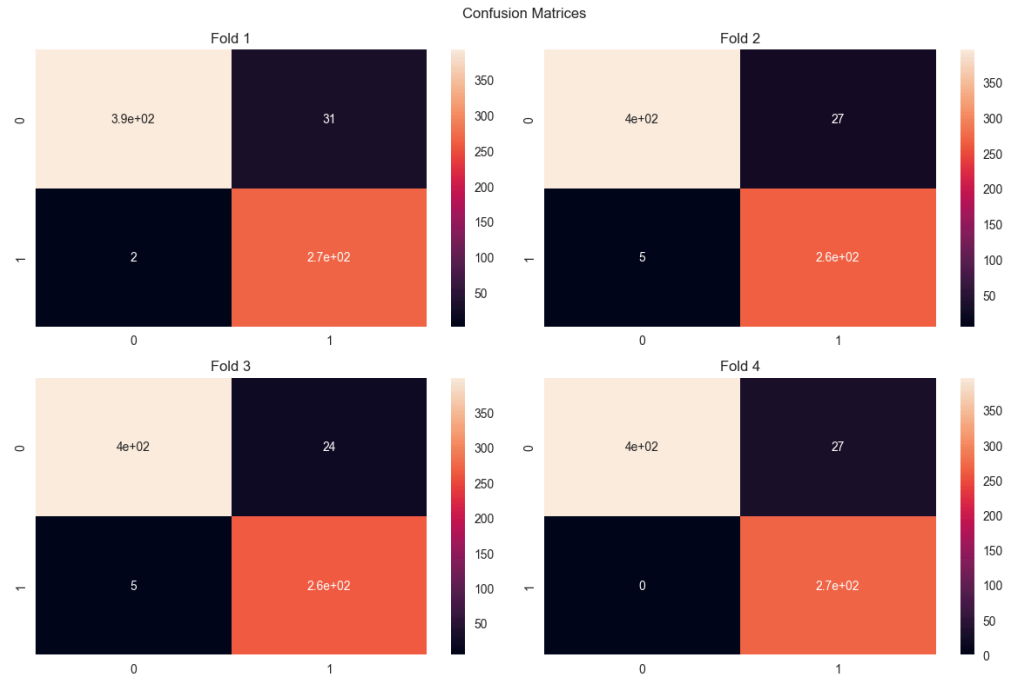


**Figure 2.1 - Confusion Matrix for TF-IDF**

## 2b GloVe

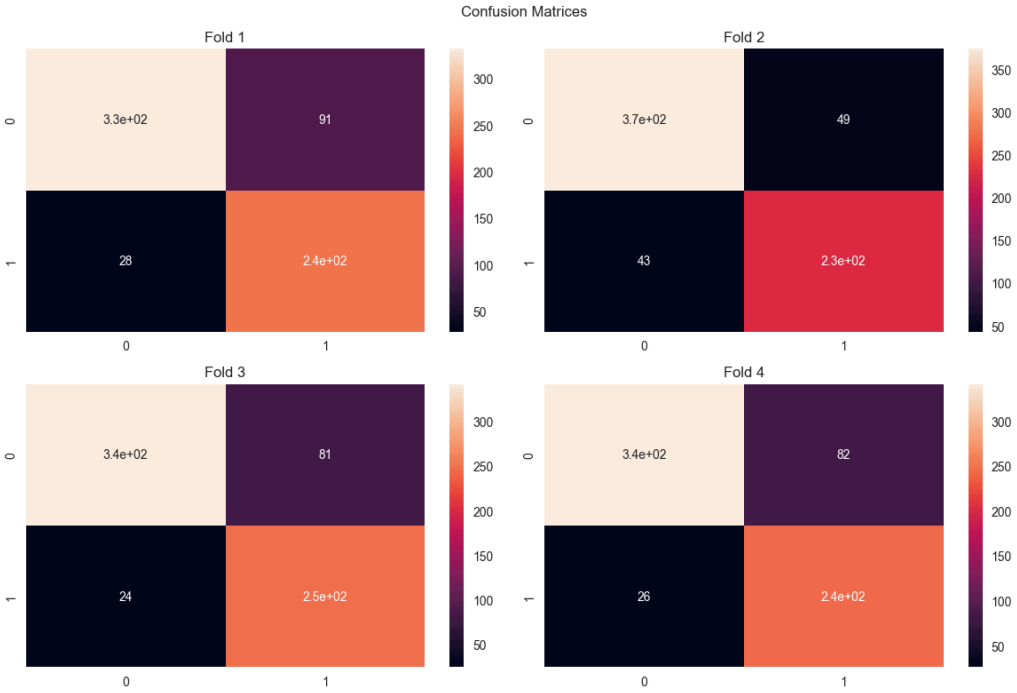


**Figure 2.1 - Confusion Matrix for GloVe**

## 2c Doc2Vec

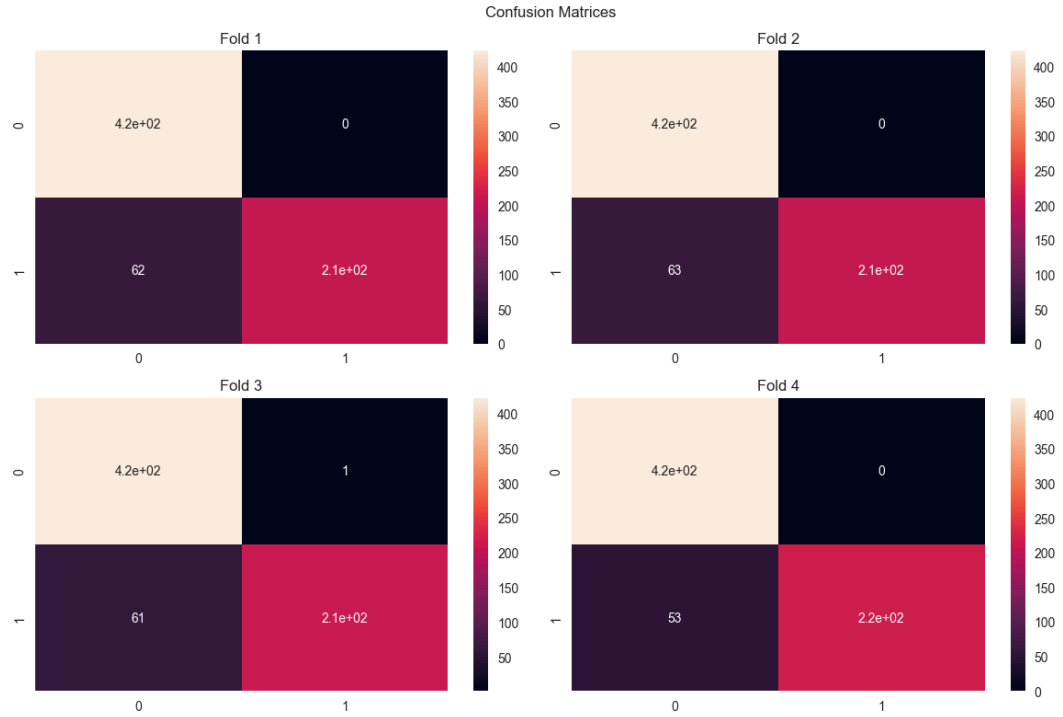


**Figure 2.1 - Confusion Matrix for Doc2Vec**

## 2d CNN

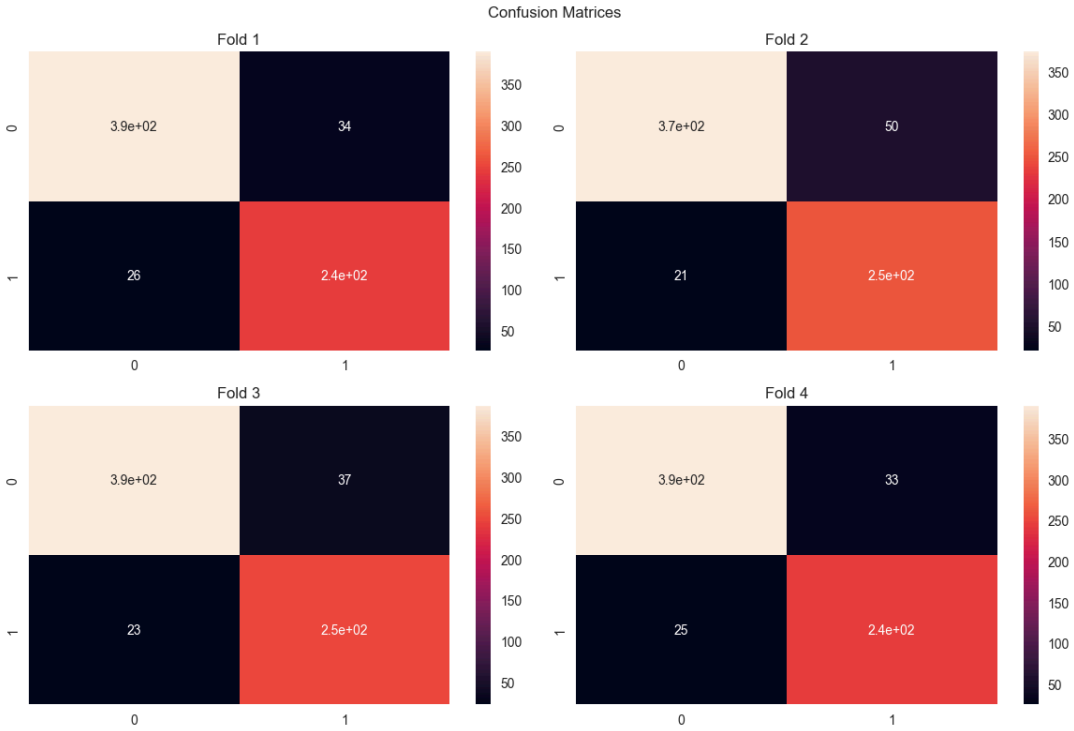


**Figure 2.1 - Confusion Matrix for CNN**

## 2e LSTM

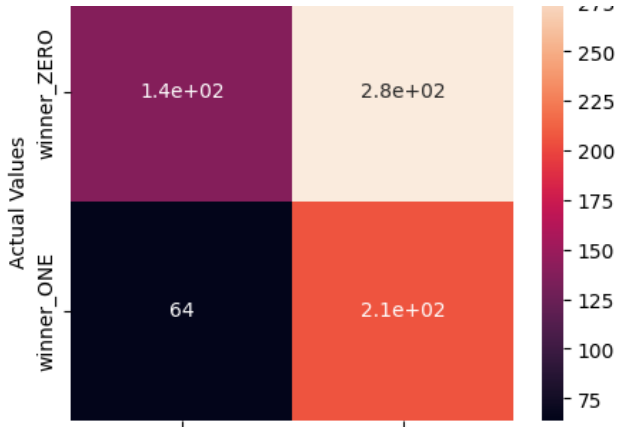


**Figure 2.1 - Confusion Matrix for CNN**

## 2f BERT

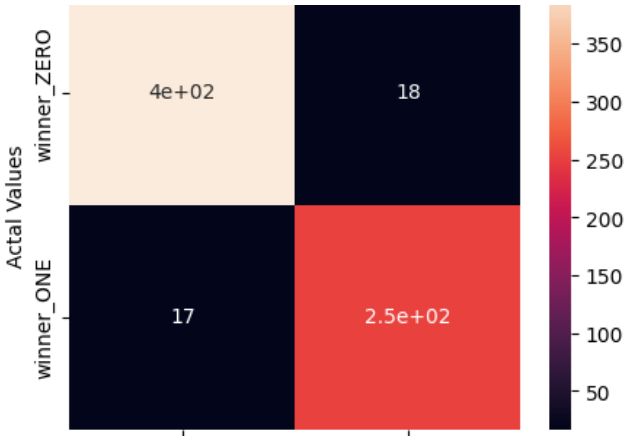


**Figure 2.1 - Confusion Matrix for CNN**

## 2g Ensemble

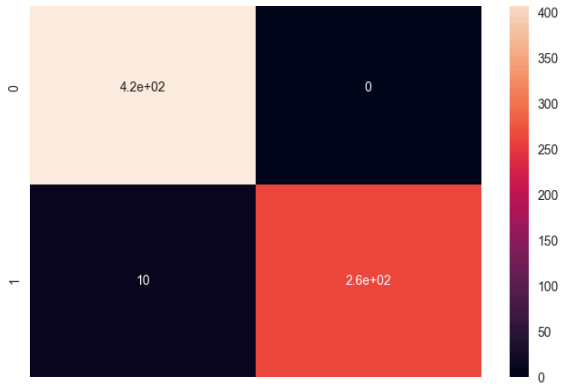


**Figure 2.1 - Confusion Matrix for CNN**

# V Summary

In this study, various supervised machine learning techniques were employed, including Support Vector Machines (SVM), decision trees and their derivatives (random forests, XGBoost, AdaBoost), KNN clustering, and Naive Bayes, to predict the ideological outcome of Supreme Court cases. The target outcome was categorized as either liberal or conservative. Overall, the models achieved a test accuracy of approximately 72%, significantly outperforming the 50% accuracy of the null model. Random forest algorithms consistently produced the best results, although other methods were also competitive. Notably, Naive Bayes performed poorly, likely due to the non-parametric nature of the categorical variables used in the study.

## 1 TF-IDF

The TF-IDF approach revealed two similar combinations: combination 3 (no preprocessing - anonymization - imbalance) and combination 4 (no preprocessing - anonymization - balanced). Each combination produced four results based on folds. The best model, the fourth of the third combination, achieved a remarkable testing accuracy of 97.2% and a testing loss of 0.141.

## 2 GloVe

The GloVe method identified the best combination as combination 2 (no preprocessing - no anonymization - balanced). The first model of this combination demonstrated the highest testing accuracy, reaching 91.6%, with a testing loss of 0.384.

## 3 Doc2Vec

For Doc2Vec, two similar combinations emerged: combination 1 (no preprocessing - no anonymization - imbalance) and combination 5 (preprocessing - no anonymization - imbalance), each with four results based on folds. The second model of the fifth combination proved to be the most effective, achieving a testing accuracy of 94.5% and a testing loss of 0.282.

## 4 CNN:

Similarly, CNN highlighted two similar combinations: combination 2 (no preprocessing - no anonymization - balanced) and combination 5 (preprocessing - no anonymization - imbalance). The second model of the second combination exhibited the highest testing accuracy, at 93.3%, with a testing loss of 0.325.

In conclusion, despite the different techniques used, the study consistently showed strong performance in predicting ideological outcomes of Supreme Court cases. Random forest algorithms were particularly effective, while Naive Bayes struggled due to the categorical nature of the variables. Ensemble methods did not significantly improve predictive accuracy, suggesting that simpler models trained on case-centered data alone were sufficient for the task.

# VI References / Bibliography

This section describes the documents and other sources from which information was gathered.

[1] Machine Learning-based SON function conflict resolution IEEE. https://ieeexplore.ieee.org/document/8969675/.

[2] Conflict Resolution Strategies In Artificial Intelligence. https://conflictresolved.com/conflict-resolution-strategies-in-artificial-intelligence/.

[3] Artificial Intelligence Techniques for Conflict Resolution. https://link.springer.com/article/10.1007/s10726-021-09738-x.

[4] Using Artificial Intelligence to provide Intelligent Dispute Resolution .... https://link.springer.com/article/10.1007/s10726-021-09734-1.